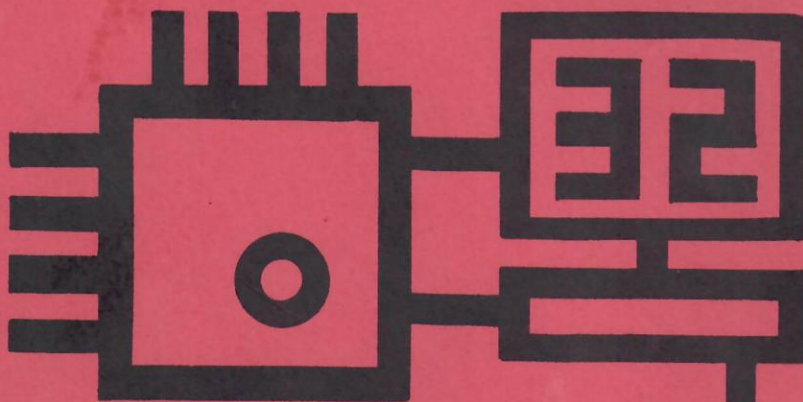




Ed-Lab

ANSWER MANUAL

ADVANCED DIGITAL SYSTEMS
ED-LAB 900



CES INDUSTRIES, INC.

REVISED ANSWERS

Ed-Lab 900 Answer Manual

FOR REVISED EXPERIMENT NO. 8

TO BE USED WITH Ed-Lab 900
Experiment Manual
"ADVANCED DIGITAL SYSTEMS"
with Copyright date 1981 or later.*

- * Experiment No. 8 is expanded to include the measurement of system waveforms.

CES INDUSTRIES, INC.
130 CENTRAL AVE.
FARMINGDALE, N.Y. 11735

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

THEORY OF THE EARTH

AUTOMATIC DIGITAL COMPUTERINSTRUCTORS NOTE:

This experiment outlines the architecture of the Automatic Digital Computer and details the function of each section of the Computer Schematic. Also the Timing sequence of the operations of the computer and the waveforms expected at key points in the schematics are given.

The timing logic is wired and the start button is operated. The student observes the computer going from OFF to ON (and back to OFF in ONE STEP). He also observes the FETCH/EXECUTE cycles on Lamps 3 and 4. He also tests the AUTO/ONE STEP operation and the STOP operation in AUTO mode.

The Instruction Logic is then constructed and tested.

The student could write a test sheet verifying operation of the Timing Logic. The test sheet could appear as shown below:

PROCEDURE	ITEM	CHECK
3.	OFF / INST LAMPS Lit	✓
4.	ONE CYCLE	✓
7.	AUTO/RUN	✓
8.	AUTO/HALT	✓

APPENDIX A

Table 1

This table lists the results of the analysis of the data collected during the study. The results are presented in the following order: the number of cases, the number of cases with a specific characteristic, the number of cases with a specific characteristic and a specific outcome, and the number of cases with a specific characteristic and a specific outcome and a specific outcome.

The results of the analysis are presented in the following order: the number of cases, the number of cases with a specific characteristic, the number of cases with a specific characteristic and a specific outcome, and the number of cases with a specific characteristic and a specific outcome and a specific outcome.

The results of the analysis are presented in the following order: the number of cases, the number of cases with a specific characteristic, the number of cases with a specific characteristic and a specific outcome, and the number of cases with a specific characteristic and a specific outcome and a specific outcome.

The results of the analysis are presented in the following order: the number of cases, the number of cases with a specific characteristic, the number of cases with a specific characteristic and a specific outcome, and the number of cases with a specific characteristic and a specific outcome and a specific outcome.

Case	Characteristic	Outcome
1	10-10-10	10-10-10
2	10-10-10	10-10-10
3	10-10-10	10-10-10
4	10-10-10	10-10-10
5	10-10-10	10-10-10

TABLE 9(d) (TIMING LOGIC WAVEFORMS)

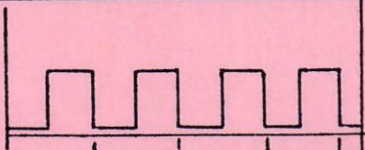
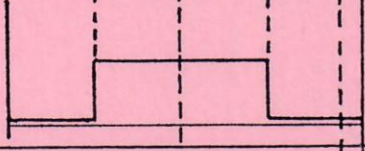
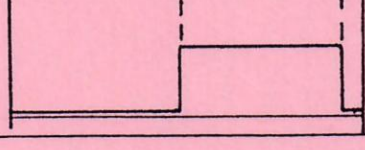
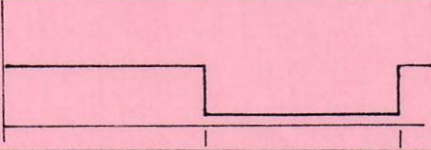
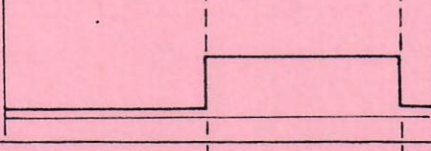
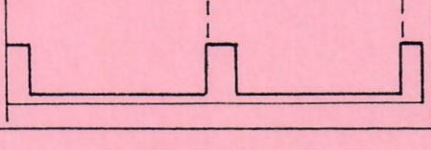
Test Point	Sketch Waveform	Check Figure
Flip-Flop 10, Q		8-2 "PULSE" Check <u>✓</u>
Flip-Flop #1, Q		8-2 "A1" Check <u>✓</u>
Flip-Flop #2, Q		8-2 "B1" Check <u>✓</u>

TABLE 9(f) (TIMING LOGIC WAVEFORMS)

TEST POINT	SKETCH WAVEFORM	CHECK FIGURE
Flip-Flop #8, $\bar{Q}$		8-3 "INST" Check <u>✓</u>
Flip-Flop #8, Q		8-3 "OP" Check <u>✓</u>
NOR #1		8-3 "BIT 0" Check <u>✓</u>

10-10-10
 10-10-10

10-10-10 10-10-10

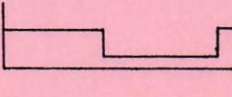
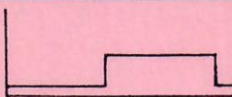
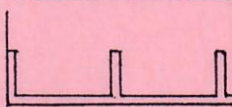
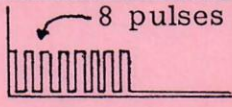
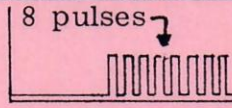
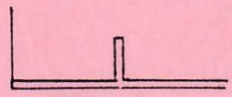
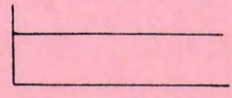
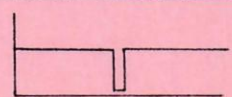
10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10

10-10-10 10-10-10

10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10
10-10-10	10-10-10	10-10-10

AUTOMATIC DIGITAL COMPUTER

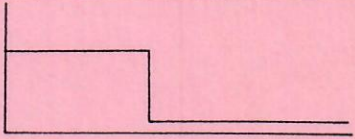
TABLE 12(e) (INSTRUCTION LOGIC WAVEFORMS)

TEST POINT (Figure 8-1B)	SET SWITCHES G H J (INSTRUCTION)	SKETCH WAVEFORM	CHECK Figure 8-5
Flip-Flop #8 $\bar{Q}$ (T1)	1 1 0 (LOAD)		"INST" ① Check ☒
Flip-Flop #8 Q (T2)	SAME AS ABOVE		"Op" ② Check ☒
NOR #1 Output (T3)	SAME AS ABOVE		"BIT 0" ③ Check ☒
INSTRUCTION Reg. SHIFT PULSES (T4)	SAME AS ABOVE		④ Check ☒
ARITHMETIC Reg. SHIFT and BIT DELAY. T (T5)	SAME AS ABOVE		⑤ Check ☒
ARITHMETIC Reg. SHIFT and BIT DELAY T (T5)	0 1 1 (SHIFT RIGHT)		⑥ Check ☒
PROGRAM CTR. SET (T6)	0 1 1 (NO TRANSFER)		⑦ Check ☒
PROGRAM CTR. SET (T6)	0 0 1 (TRANSFER)		⑧ Check ☒

TEST POINT (Figure 8-1B)	SET SWITCHES G H J (INSTRUCTION)	SKETCH WAVEFORM	CHECK Figure 8-5
PROGRAM CTR SET (T6)	0 0 0 (TRANSFER ON NEGATIVE) (Ground A Reg. DATA IN)*		⑨ NO NEGATIVE NUMBER Check ☒
PROGRAM CTR SET (T6)	0 0 0 (TRANSFER ON NEGATIVE)(Remove Ground to A Reg. DATA IN)*		⑩ NEGATIVE NUMBER Check ☒
MEMORY READ/WRITE (T7)	1 1 0 (LOAD)		⑪ Check ☒
MEMORY READ/WRITE (T7)	0 1 0 (STORE)	8 pulses →	⑫ Check ☒
BIT DELAY NS (T8)	1 0 1 (ADD)		⑬ Check ☒
BIT DELAY NS (T8)	1 0 0 (SUBTRACT)		⑭ Check ☒
NOT #1 Output (T9)	1 0 1 (ADD)		⑮ Check ☒
NOT #1 Output (T9)	1 0 0 (SUBTRACT)		⑯ Check ☒
FULL ADDER EA (T10)	0 1 0 (STORE)		⑰ Check ☒

* When DATA IN on A. REG. is not grounded the A. REG. will contain all "1"s and the TRANSFER ON NEGATIVE instruction will detect a NEGATIVE NUMBER.

AUTOMATIC DIGITAL COMPUTER

TEST POINT (Figure 8-1B)	SET SWITCHES G H J (INSTRUCTION)	SKETCH WAVEFORM	CHECK Figure 8-5
FULL ADDER EA (T10)	1 1 0 (LOAD)		⑮ Check ☒
OBSERVE L3 and L4	1 1 1 (HALT)	COMPUTER TURNS OFF L3 = ON L4 = OFF	NONE Check ☒

STATE OF NEW YORK

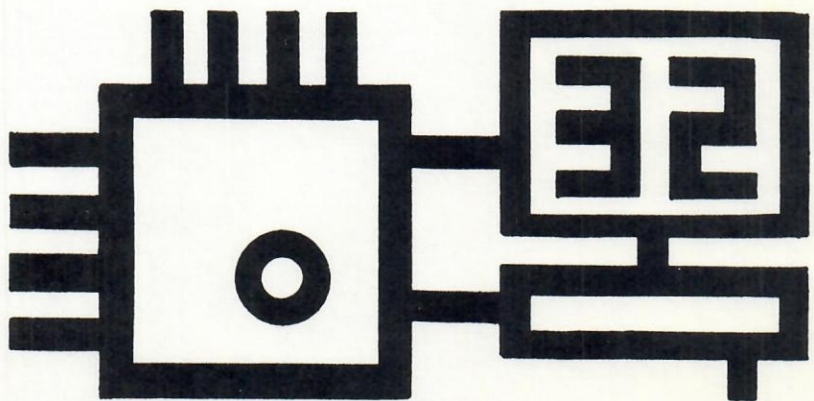
COUNTY OF SARATOGA	TOWN OF SARATOGA	SCHOOL DISTRICT (SARATOGA)	SCHOOL DISTRICT (SARATOGA)
SARATOGA SARATOGA	SARATOGA SARATOGA	SARATOGA SARATOGA	SARATOGA SARATOGA
SARATOGA SARATOGA	SARATOGA SARATOGA	SARATOGA SARATOGA	SARATOGA SARATOGA



Ed-Lab

ANSWER MANUAL

ADVANCED DIGITAL SYSTEMS
ED-LAB 900



CES INDUSTRIES, INC.



Ed-Lab ANSWER MANUAL

ADVANCED DIGITAL SYSTEMS
ED-LAB 300



SERIAL NUMBER _____

INSTRUCTOR'S NAME _____

Please notify CES when transferring
Answer Manual

ED - LAB 900

INSTRUCTOR'S MANUAL

DIGITAL TECHNOLOGY

COPYRIGHT © 1973, 1978 CES INDUSTRIES, INC.

ALL RIGHTS RESERVED

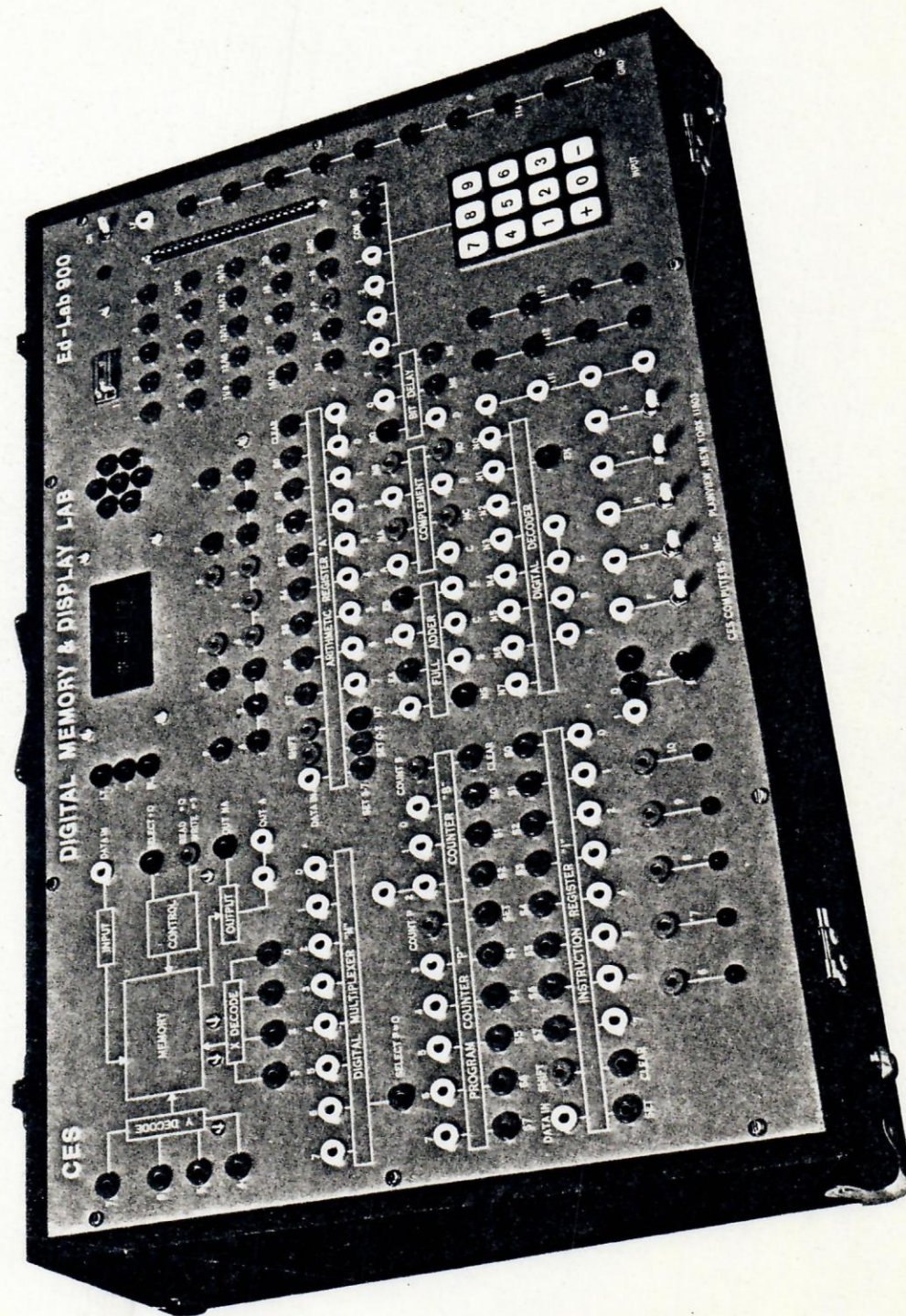
CES INDUSTRIES, INC.
ED-LAB EDUCATIONAL PRODUCTS SYSTEMS
130 CENTRAL AVENUE
FARMINGDALE, NEW YORK 11735
(516) 293-1420

SERIAL NUMBER
INSTRUCTOR'S NAME
Please call CES when transferring Address Name

ED - LAB 200
INSTRUCTOR'S MANUAL
DIGITAL TECHNOLOGY

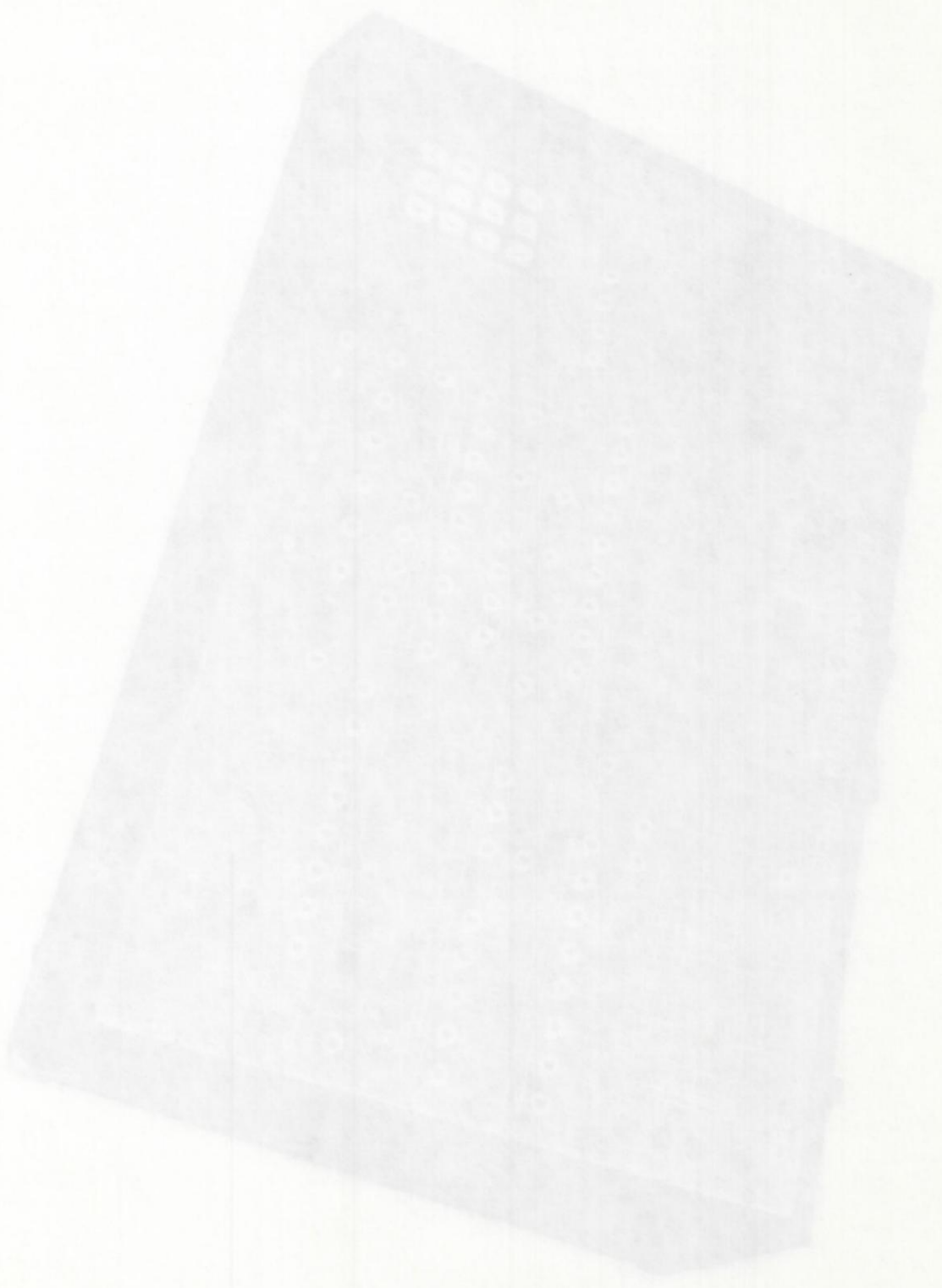
COPYRIGHT © 1975, WESTER INDUSTRIES, INC.
ALL RIGHTS RESERVED

WESTER INDUSTRIES, INC.
ED-LAB INTERNATIONAL PRODUCTS SYSTEMS
100 CENTRAL AVENUE
CARLETON PLACE, NEW YORK 10020
(212) 224-1234



ED-LAB 900 SUITCASE MODEL

FROM JACQUES FOR JAL-22



ED-LAB 900 ANSWER MANUAL

TABLE OF CONTENTS

CHANGE OF ADDRESS NOTICE

- I PREFACE
- II ED-LAB 900 EXPERIMENT LIST
- III ERRATA SHEETS
- IV ADD-ONS: ED-LAB 101, ED-LAB 380 and ED-LAB 650
- V ANSWERS TO EXPERIMENTS

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

ED-1AS FOR ADDRESS LIST

I. PREFACE

The Ed-Lab 900 Digital Input/Output/Memory/Computer Trainer is an add-on to the Ed-Lab 700 Digital Logic Trainer. The Ed-Lab 900 contains a Keyboard, Registers, Display, Multiplexer, Adder, Counters, Memory and other elements to implement experiments in a variety of Digital Systems and Applications.

Another ADD-ON to the Ed-Lab 700 and Ed-Lab 900 is the Ed-Lab 101 which includes an Integrated Circuit Breadboard mounted on a circuit card that plugs in to the sockets on the Ed-Lab 700 and Ed-Lab 900. A new ADD-ON is the Ed-Lab 380 Microprocessor which is used in conjunction with the Ed-Lab 900.

The first edition of the Ed-Lab 900 Lab Manual in 1973 had 20 chapters and the revised edition in 1977 contains 25 chapters, including a total of 87 experiments. If you are using the earlier 1973 edition, contact CES for the additional experiments.

Chapters 1 thru 7 (including experiments 7-1 thru 7-6) are performed using the Ed-Lab 900 alone. Experiments 7-7 thru 7-9 and all other experiments use the Ed-Lab 900 and the Ed-Lab 700 together. The experiments using both trainers cover such systems as: Digital Computers, Programming, Integrated Circuits, Numerical Control, Electronic Music, Digital Communications, Digital Voltmeter, Display Systems, and Digital Frequency Synthesizer. A list of all experiments in the manual follows.

The 1977 Edition has the following experiments in addition to those in the 1973 edition: 4-3, 7-2(revised) 7-6 thru 7-9; and Chapters 21 thru 25. Experiment No. 7 covers the Adder plus Octal Arithmetic. The Addition and Subtraction of Positive and Negative Numbers using Complements is difficult and the instructor may permit skipping some of the student examples which follow Experiment No. 7. These can be completed by the student later. Experiment No. 12 requires that the student develop and prepare his own programs. This is also difficult. The instructor may make these optional and provide the programs to the students.

The experiments can be separated into 7 overall groups:

1. Operation of Ed-Lab 900 Logic Elements (Experiments 1 to 7-1)
2. Arithmetic (Experiments 7-2 to 7-9)
3. Digital Computer and Programming (Experiments 8 to 12, 19)
4. Other Digital Systems (Experiments 14 to 18, 21 to 25)
5. Digital Instrument (Experiments 20-1 to 20-3)
6. Input-Output Terminals (Experiments 20-4 to 20-6)
7. Integrated Circuits (Experiment 13)

The student should cover Module 1 above to gain familiarity with the trainer. It then remains the option of the teacher to require that the student follow experiments in sequence, or to allow the student to select further experiments at his own option.

Reference Texts include:

"Digital Principles and Applications" McGraw-Hill; by A. Malvino & D. Leach.
"Boolean Algebra with Computer Applications" McGraw-Hill; by G. Williams.
"Digital Computer Fundamentals" McGraw-Hill; by T. Bartee.
"Digital Computer Principles" McGraw-Hill; by Burroughs Corp.
"Basics of Digital Computers" Rider; by J. Murphy.
"Digital Logic Circuits" Hayden; by S. Libes
"Understanding Digital Computers" Hayden; by R. Benrey
"Digital Logic and Computer Operations" McGraw-Hill; by Baron & Piccirilli
"Computers for Technicians" Prentice Hall; by A. Marcus & J.D. Lenk
"The TTL Data Book" Texas Instruments; by T.I. Staff
"Digital Electronic Circuits" Parker Brothers; by F.W. Hughes

Instructors have the option of using the Lab Manual exactly as it is and requiring that the student follow experiments in the sequence presented or he may select experiments and text references and list a sequence of requirements for the student to follow. This varies with the level of the course given and the time available. A typical assignment list or self-study list is given below (there are no Text A or B as referenced, these are only format examples.)

Module: Counters, Register & Keyboards; Module M1

TEXT A, page 2 - 18
TEXT C, page 34 - 40
LAB (900) Exp. 1 (submit REPORT)
LAB (900) Exp. 2-1 to 2-4 (submit REPORT)
INSTRUCTOR CONFERENCE C1
TEXT C page 42-45
LAB (900) Exp. 2-5 to 2-7 (submit REPORT)
NOTES M1 and M2 (submit ANSWERS to Quiz M1 & M2)
LAB (900) Exp. 3 (submit REPORT)
LECTURE L1 (Octal Numbers)
LAB (900) Exp. 4-1 and 4-2 (submit REPORT)
LAB (900) Exp. 4-3 (REPORT optional)
AUDIO VISUAL AV3 (Intro to computers)
TEXT A page 19 - 25 (optional reference)
INSTRUCTOR CONFERENCE C2
WRITTEN TEST WT1
PRACTICAL TEST PT2
.....END OF MODULE M2

Note that the above listing would be most suitable to a self-study program. The texts would be selected by the instructor. The tests would reflect the level of proficiency desired by the program. Another format would be the separate LECTURE-LAB format, and the instructor would then cover the Lectures, Theory, Tests, and Conferences in the conventional LECTURE part, and the LAB would only require experiments and LAB REPORT submission.

CES provides a service to instructors where we will be pleased to support your programs and answer any technical or other questions that you may have. We will also support the up-keep of your equipment should any malfunction occur. We appreciate all inquires. Please write to:

CES INDUSTRIES, INC.
130 Central Avenue
Farmingdale, N.Y. 11735

or call: (516) 293-1420

II. ED-LAB 900 EXPERIMENTS

1. INTRODUCTION & COUNTERS

- Introduction
- 1-1. Bit Counter
- 1-2. Pre-Set Counter
- 1-3. Cascaded Counter
- Trainer Summary

2. DISPLAYS, REGISTER AND OCTAL CODING

- 2-1. Seven Segment Display
- 2-2. Binary Coded Decimal Display
- 2-3. Shift Register
- 2-4. BCD Register Display
- 2-5. Octal Register Display
- 2-6. Octal Fraction Display
- 2-7. BCD Magnitude and Sign Display
- 2-8. Segment "a" Logic

3. KEYBOARD DATA INPUT

- 3-1. Keyboard BCD Coding
- 3-2. Keyboard to Display
- 3-3. Data Entry and Octal Display
- 3-4. BCD Display, Keyboard and Register

4. RANDOM ACCESS MEMORY

- 4-1. Random Access: Write-Read
- 4-2. Memory, Register and Display
- 4-3. Automatic Address Memory

5. REGISTER PRESETS AND DECODING

- 5-1. Register and Decoder

6. MULTIPLEXING AND DATA TRANSFER

- 6-1. Preset Instruction Register
- 6-2. Counter Preset & Data Transfer
- 6-3. Multiplexer

7. ADDER AND OCTAL ARITHMETIC

- 7-1. Full Adder
- 7-2. Adder with Memory
- 7-3. Subtraction with Memory
- Octal Arithmetic
- 7-4. Bit Delay Master Slave
- 7-5. Master - Slave Triggering
- 7-6. Serial Register and Adder
- 7-7. Instruction Arithmetic Unit
- 7-8. Programmed Memory and Arithmetic Unit
- 7-9. Line Frequency Counter

8. AUTOMATIC DIGITAL COMPUTER

Block Diagram
Operations
Program Instructions

- 8-1. Timing & Control
- 8-2. Arithmetic Unit
Logic Troubleshooting
- 8-3. Memory and Address Control
- 8-4. Automatic Digital Computer

9. ARITHMETIC AND MEMORY SYSTEM

- 9-1. Arithmetic Unit
Instruction Tables
- 9-2. Memory Unit
- 9-3. Test Maintenance Programs
Troubleshooting

10. PROGRAMMING AND FLOW CHARTS

- Programs
Symbolic Program
- 10-1. Machine Language
- 10-2. Flow Charts

11. ITERATIVE PROGRAM LOOPS

- 11-1. Multiplication
- 11-2. Iterative Loops
- 11-3. Decision Loops

12. PROGRAMMING TECHNIQUES

- 12-1. Command Variations
- 12-2. Bit Selection
- 12-3. BCD to Binary Conversion
- 12-4. Binary to BCD Conversion
Decimal Computation
- 12-5. Overflow
Subroutines
Computer Languages

13. PRINTED CIRCUITS AND INTEGRATED CIRCUITS

- 13-1. Integrated Circuits
P.C. Card

14. DIGITAL OUTPUT REFRESHER

- 14-1. Data Refresher

15. DIGITAL COMMUNICATIONS

- Pulse Code Modulation
- 15-1. Transmit Data Register
- Pulse Amplitude Modulation
- Frequency Shift Keyer
- PDM Modulator
- 15-2. PDM Modulator - Transmitter
- 15-3. PDM Decoder
- PDM Modem
- 15-4. Remote Display System

16. NUMERICAL CONTROL SYSTEM

- 16-1. Sequential Memory
- 16-2. Numerical Control Graphics Display

17. ELECTRONIC MUSIC

- 17-1. Tone and Octave Generation
- 17-2. Digital Sequence Control
- 17-3. Keying System
- Rhythm Counter
- 17-4. Rhythm System
- 17-5. Base Rhythm
- 17-6. Tuning
- 17-7. Rhythm Controller

18. DIGITAL MODEM

- Digital Pulse Width Modulation
- Digital Phase Modulation
- Synchronization
- ASCII 8 Bit Code
- ASCII Display Connection
- 18-1. Phase Modulation Transmitter
- 18-2. Phase Modulation Receiver
- PCM Carrier Frequency Modulator
- PCM Detector
- Photocell Signal Conditioner

19. ADDITIONAL PROGRAMMING METHODS

- Retrieval
- Location
- Search
- Sort
- 19-1. Location Problem
- 19-2. Scaling
- 19-3. Double Precision
- Floating Points
- 19-4. Real-Time Timing
- 19-5. Square - Root Algorithm
- 19-6. Problem Solving

20. INPUT-OUTPUT INTERFACE

- Digital Voltmeter Display
- 20-1. 0.5 Volt Resolution
- 20-2. 0.25 Volt Resolution
- 20-3. Digital Voltmeter Instrument
 - Computer Peripheral Equipment
 - Remote Terminal
- 20-4. Remote Output Display Terminal
- 20-5. Remote Input - Output Terminal
 - Interactive Terminal
- 20-6. Calculator Program

21. ALPHANUMERIC DISPLAY SYSTEM

- 21-1. Alphanumeric CRT Display

22. GRAPHICS CRT DISPLAY

- 22-1. Graphic Format Display

23. MORSE CODE KEYER

- 23-1. Code Keyer Logic

24. AUTOMATIC KEYBOARD ENTRY

- 24-1. Automatic Keyboard Entry

25. DIGITAL FREQUENCY SYNTHESIZER

- 25-1. Digital Frequency Synthesizer
- 25-2. Phase Comparator
- 25-3. Frequency Comparator
- 25-4. VCO Oscillator
- 25-5. Phase-Locked-Loop/open loop
- 25-6. Phase-Locked-Loop/closed loop

ERRATA, CLARIFICATIONS & ADDITIONS

CES ED-LAB 900 MANUAL
(Edition Dated 1977)

CES INDUSTRIES, INC.
130 Central Avenue
Farmingdale, New York 11735
Phone: 516-293-1420

BRASS, CLAMPING AND LOCKING

FOR THE USE OF THE
(Edition 2000 1977)

ONE INDUSTRIAL, INC.
130 Central Avenue
Beverly Hills, New York 11590
Phone: 718-333-1410

ERRATA, CLARIFICATIONS AND ADDITIONS
III. ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

(Issued April 6, 1978)

These pages on ERRATA, CLARIFICATIONS AND ADDITIONS include corrections, clarifications, and some additions or improvements to the original text.

EXP. No. 1

No Errata

EXP. No. 2

Page 12: Question 6, 2nd line change "decimal coded" to "coded decimal"

EXP. No. 3

Page 2: Nine lines from bottom: Remove "(The negative pulse is too short to be seen on the lamp.)"
and Add in its place the following three paragraphs:

"There is a "0" (LAMP OFF) on Lamp 6 when key is depressed.

What occurs on Lamp 6 when any key is depressed? _____

Remove Lamp 6 wire from "S" output from the keyboard. Connect Lamp 6 to DS or Delayed Strobe output. Depress any key. The "0" pulse (LAMP OFF) is too short to be seen on the Lamp. If the DS output is connected through an inverter (DS to COMPLEMENT A input: Output NA to Lamp 6), then a short positive pulse "1" can be observed with the eye."

EXP. No. 4

Page 6: Remove Paragraph from bottom of page, Starting with:
"DESCRIPTION (AUTOMATIC ADDRESS) In the previous"....
and ending with "is DOWN"

Page 9: In TABLE FOR CES PHONE NO. change "BIT ADDRESS (H,J,K)" to
"BIT ADDRESS: SWITCHES (H,J,K); MEMORY (2,1,0)".

In TABLE column marked ADDRESS; change "OCTAL (H,J,K)" to
"MEMORY INPUTS (6,5,4)"

EXP. No. 5

No Errata

EXP. No. 6

Page 4: Line 13, change "SET input load to "SET input lead".

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 7

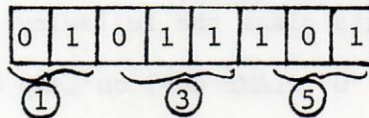
Page 1: In diagram 7-1 in Full Adder block change "AE" to "EA" and change "BE" to "EB"

Page 3: In Table B, Column Marked "DECIMAL"
Place "not used" on line corresponding to "C" and
Place "not used" on line corresponding to (K) CARRY

Page 4: Add notations to Table C next to EB = 0 of "C = 0"
Add notation to Table D next to EA = 0 of "C = 0"

Page 8: Paragraph 2, line 2 change "SWITCH J UP then DOWN" to "SWITCH J DOWN then UP"

Page 9: Diagram on top of Page Change circled "2" to "1"



Page 14: In Diagram 7-6, On BIT DELAY, change " $\overline{\text{NR}}$ " to "NR"

Page 18: In Diagram 7-7, On BIT DELAY, change " $\overline{\text{NR}}$ " to "NR"

Page 19: Table, column marked DATA "B", change DATA "B" to "DATA "B" (SWITCH F)

Page 20: In column marked DISPLAY change "072" to "166"
change "041" to "135"

Page 22: In diagram 7-8, On BIT DELAY, change " $\overline{\text{NR}}$ " to "NR"

Page 25: Step 4, line two change "Keep Switch D DOWN (for reading)" to
"Keep Switch C DOWN (for reading)"

Page 27: In Diagram 7-9 On BIT DELAY, change " $\overline{\text{NR}}$ " to "NR"

Page 27: Third line from the bottom of page, change "A REGISTER" to "PROGRAM COUNTER".

Page 28: Step 2, line 1: change "Register" to "Counter"

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 8

Page 11: On the Ed-Lab 700 LAMP LABELS
change:

ONE STEP	to	AUTO
AUTO		ONE STEP

On the Ed-Lab 900 SWITCHES LABELS
change:

MANUAL SET	to	AUTO
AUTO		MANUAL SET

EXP. No. 9

Page 7: Step 10-6

Line 3 : Add the following note after "Pulses only on
STORE 010"

NOTE: SYNC ON OPERATE SIGNAL, LAMP 2 of Ed-Lab 700

Line 4: Change "TRANSFER (NAND #2)" to "TRANSFER (NAND #1)"

Line 8: Change "INST (FF #8 Q)" to "INST (FF #8 $\bar{Q}$)"

Paragraph 9.12 3rd line change "the address always
shows the address, one" to "the address lamps always
indicate the address location one".

Page 12: Change program steps 15, 16, 17, 20 to read as follows:

15	11(4)	107	
16	11(0)	107	TRANS NEG (0)
17	21(5)	122	
20	21(7)	135	HALT (7)

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 10

page 6: Table at top of page (OPERATING INSTRUCTIONS)(in Block marked NOTES)

change:	05 for A + B	to	04 for A + B*
	11 for A - B	to	10 for A - B
	15 for B - A	to	14 for B - A
	21 for 2A- B	to	20 for 2A- B

Add note:

*Note: Transfer Instruction
actually transfers to
address +1; see Exp. 9
page 1.

page 9: Make changes in Column marked "COMMENTS" (MAGNITUDE SIGN)

at address 07; change "STORE 1A1 in 25" to "STORE |A| in 25"
" " 12; change "SUB A results in 1A1" to "SUB A results in |A|"
" " 16; change "ADD 1A1 from 25" to "ADD |A| from 25"
" " 22; change "SUB B results in 1B1" to "SUB B results in |B|"

EXP. No. 11

Page 6: At Location 15 on chart, in INST COLUMN,
change "TRN" to "TON"

EXP. No. 12

Page 1: In DESCRIPTION line 7 from top change "less different" to "fewer".

Page 6: 2nd line from bottom change "(4 x 2) = 16" to "(4 x 2) = 8"

Page 7: In Table of BCD to BINARY, column marked BINARY (OCTAL)
change "031" to "061"

Page 9: Insert the following note on the side of the table entitled
OVERFLOW TRUTH TABLE

"A & B & SUM refer to the MOST SIGNIFICANT
DIGITS of the TWO NUMBERS BEING PROCESSED.

Page 10: In "MAG" ROUTINE = |A| Table

6th line from top in Table change "A SUB" to "1A SUB"

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 12

Page 11: Line 1: Change "A LOAD L1 thru L9 ..." to
"L17 LOAD L1 thru L7..."

Line 2: Change "L5 STORE" to "LA STORE"

Between Line 3 and 4 (i.e. after L11 LOAD)

add "L6 STORE STORE EXIT INSTRUCTION IN SUBROUTINE"

Line 8: Change "C LOAD" to "L19 LOAD"

Line 9: Change "L5 STORE" to "LA STORE"

Between Lines 10 and 11 (i.e. after L16 LOAD).

add "L6 STORE STORE EXIT TRANSFER INTO SUBROUTINE"

Between Lines 17 and 18 (i.e. after L15 STORE)

add "HALT"

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 13

Page 3: Table, INPUT Column, "H" column, 2nd line from bottom change "1" to "0"

EXP. No. 14

Page 2: Schematic Diagram, Experiment 14-1 DIGITAL OUTPUT REFRESHER
Add the following note:

Note: The D flip-flop with the D input connected to the 0 output of the INSTRUCTION REGISTER is labeled BIT DELAY and is the BIT DELAY Flip Flop on the Ed-Lab 900.

The two flip flops used as ONE SHOTS in the lower right of the diagram are located on the Ed-Lab 700.

EXP. No. 15

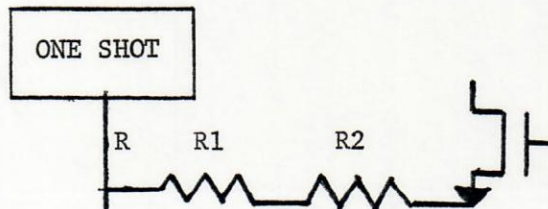
Page 1: In OBJECTIVE, 3rd line change "transmission" to "transmitter"

Page 8: Experiment 15-2 PDM MODULATOR TRANSMITTER

Schematic Diagram

(Resistors R1 and R2 in series replace the wire connections from the Transistor Emitter to the R input of the ONE-SHOT)

Change Schematic as shown below:



Page 10: Experiment 15-3 PDM DECODER SCHEMATIC

Replace "+R" at top of X POT with "1"

Replace "+R" at input to Threshold Logic T with "1".

Add note: "1 indicates connection to TI BUS (Reference Voltage)"

EXP. No. 16

Page 2: line 13 from top change "The OFF setting" to "When Flip-Flop #9 changes from ON to OFF, the Q"

Page 3: Experiment Logic 16-1 SEQUENTIAL MEMORY.

On the ADDER change: "BE" to "EB" and change "AE" to "EA"

Page 7: At the end of the first paragraph, add the following sentence:
"Temporarily disconnect the clock from Flip-Flop #7."

In 2nd paragraph, 2nd line, between "as in Figure 16-1)" and "Fast read-out" add the following note:

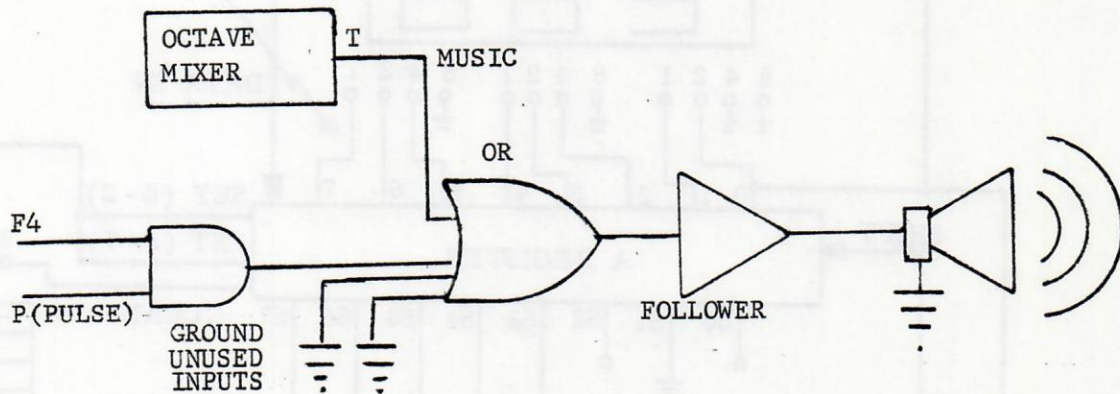
"Reconnect the clock to the Trigger of Flip-Flop #7"

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 17

Page 3: Line 7, change "(#3) acts" to "acts"

Page 11: In the diagram Experiment Logic 17-5 BASE RHYTHM Schematic:
Change "P" at output of MIXER to "T"
Remove the KEYER and the wires to and from it.
Replace with the circuit shown below:



Page 14: Experiment Logic 17-7 RHYTHM CONTROLLER. Add Note: "Add Flip-Flops #7 & #8 after counter (FLIP-FLOP #6) to extend the counter if the rhythm is too fast."

EXP. No. 18

Page 6: 3rd line below PHASE COUNTER table, change "CLOCK output" to "RESET input"

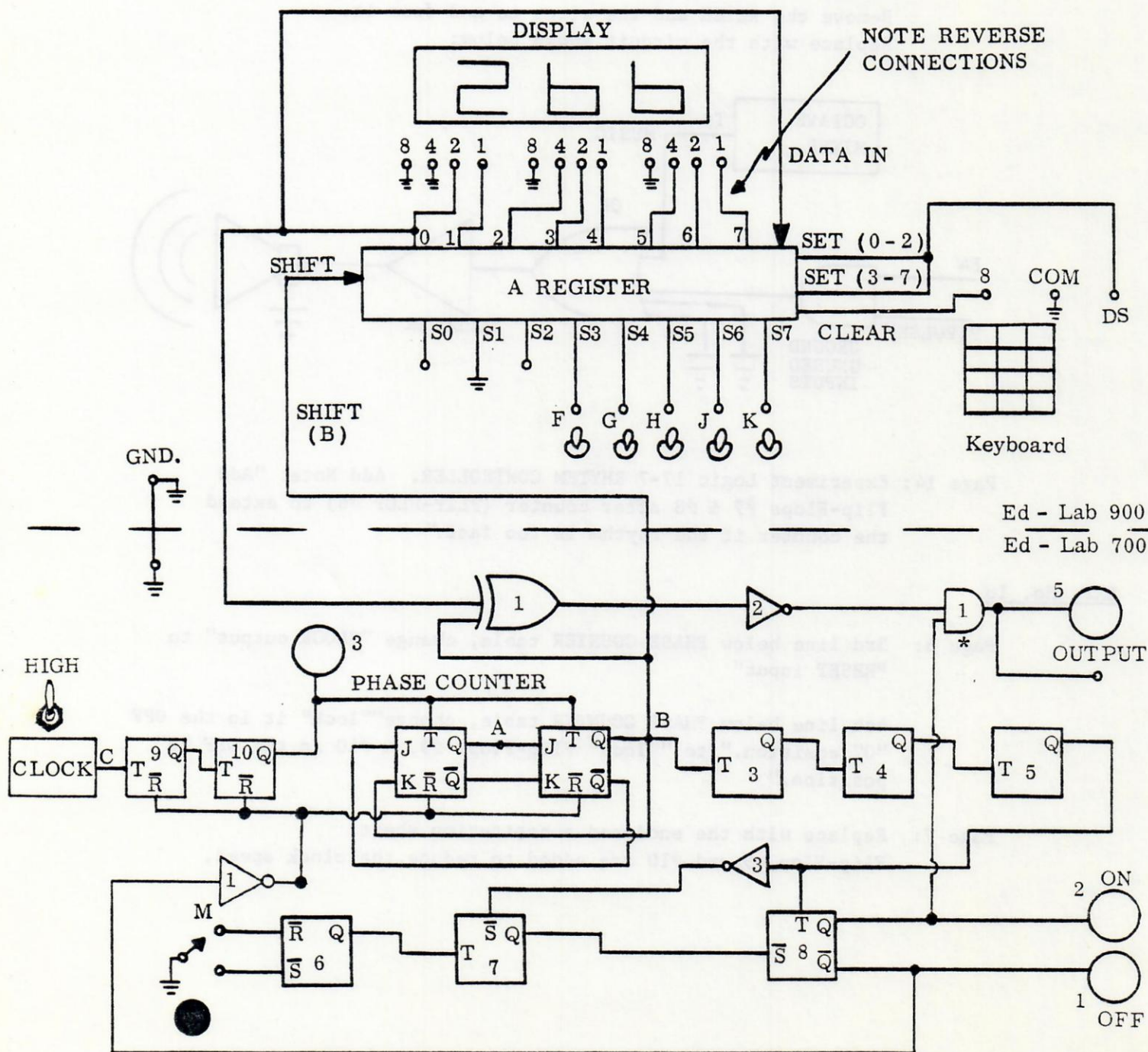
4th line below PHASE COUNTER table, change ""lock" it in the OFF "0" position." to ""lock" Flip-Flops #9, & #10 in the OFF "0" position."

Page 7: Replace with the enclosed substitution sheet.
Flip-Flop #9 and #10 are added to reduce the clock speed.

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 18

Page 7:



Experiment 18 - 1

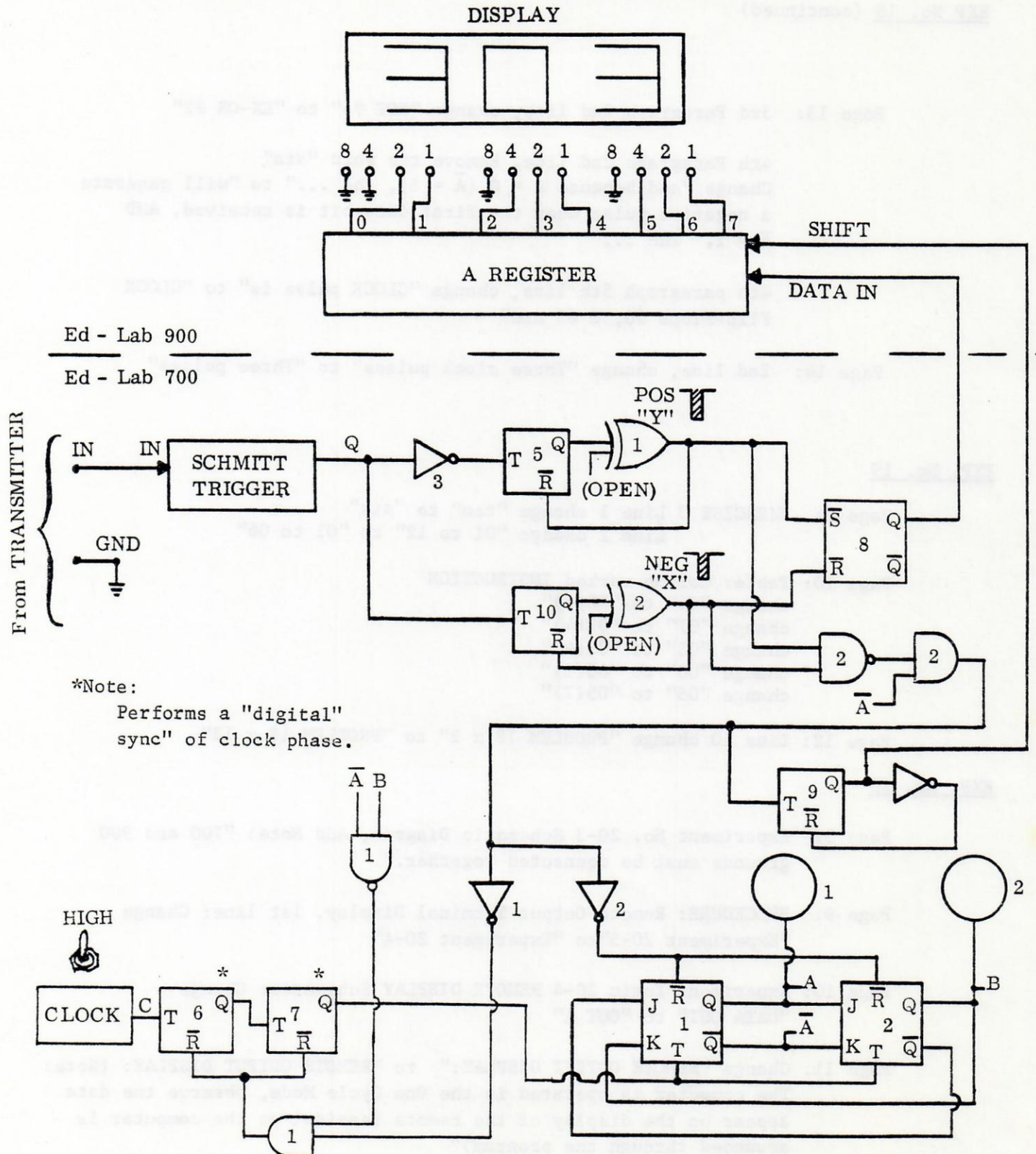
PHASE - MODULATION TRANSMITTER

*Added to prevent transmission of data when setting new data into transmitter keyboard.

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
 For Edition Dated 1977

EXP. No. 18

Page 12: Replace page 12 with the diagram below:



Experiment 18-2

PHASE - MODULATION RECEIVER

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 18 (continued)

Page 13: 3rd Paragraph 2nd line, change "NOT #5" to "EX-OR #2"

4th Paragraph 2nd line, Remove the word "Via".
Change "and because $A = 0$ ($\bar{A} = 1$), the ..." to "will generate a negative pulse when the first data bit is received, AND $\bar{A} = 1$. The ..."

4th paragraph 5th line, change "CLOCK pulse is" to "CLOCK Flip-Flops #6, & #7 are"

Page 14: 2nd line, change "Three clock pulses" to "Three pulses"

EXP. No. 19

Page 9: EXERCISE 3 Line 1 change "ten" to "six"
Line 2 change "01 to 12" to "01 to 06"

Page 10: Table: Column marked INSTRUCTION
change "06" to "06(6)"
change "07" to "07(4)"
change "03" to "03(4)"
change "00" to "00(1)"
change "05" to "05(7)"

Page 12: Line 10 change "PROBLEM 18 x 2" to "PROBLEM 18 x 12"

EXP. No. 20

Page 3: Experiment No. 20-3 Schematic Diagram, Add Note: "700 and 900 grounds must be connected together."

Page 9: PROCEDURE: Remote Output Terminal Display, 1st line: Change "Experiment 20-3" to "Experiment 20-4"

Page 10: Experiment Logic 20-4 REMOTE DISPLAY Schematic: Change "DATA OUT" to "OUT A"

Page 11: Change "REMOTE OUTPUT DISPLAY:" to "REMOTE OUTPUT DISPLAY: (Note: The computer is operated in the One Cycle Mode, observe the data appear on the display of the remote terminal as the computer is advanced through the program)"

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 20 (continued)

Page 13: Table, change "OCTAL CODE" to "ASCII CODE"

Column marked ALPHA NUMERIC, line 16, change "End of Message" to "Period"

Column marked ALPHA NUMERIC, line 17, Add "End of Message"

PROCEDURE (Remote Input-Output Computer Terminal):

Line 1 change "(20-3)" to "(20-4)"

Line 12 change "(FF #2, Q)" to "(FF #3, Q)"

Line 13 change "With a FF #2, Q = 0" to "With a FF #3, Q = 0"

Line 15 change "OR #2" to "OR #3"

Page 14: Experiment Logic 20-5 REMOTE TERMINAL Schematic - ADDER block.

Change "AE" to "EA"

Change "BE" to "EB"

Mark line from "8" of Keyboard to A Register "CLEAR" at the A Register

Page 15: Line 10, Change "AND #1" to "AND #2"

PROCEDURE (Remote Input-Output Computer Terminal):

Line 1, change "20-4" to "20-5"

Page 19: Line 8, change "Terminal Program into" to "Terminal Program (Program given on next page) into..."

Page 20: Address 12 change WORD "235" to "031"

Page 22: Line marked $X = \sqrt{017 + 012}$ add "Note: numbers in octal"

EXP. No. 21

No Errata

EXP. No. 22

No Errata

EXP. No. 23

No Errata

EXP. No. 24

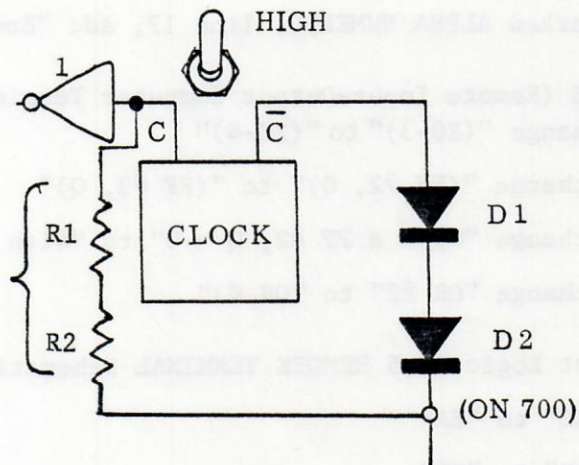
No Errata

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. No. 25

Page 3: Schematic Experiment 25-1 DIGITAL FREQUENCY SYNTHESIZER LOGIC
Add R1 and R2 and the accompanying note as shown below:

Note: Add R1
& R2 if 2000Hz
output frequency
cannot be achieved.
(see text)



Page 6: Paragraph 3, add after "can you properly adjust the clock? ____"
this note:

Note: If you are not able to reach the frequency of
2KHz, add R1 and R2 shown in Figure 25-1.

Page 7: Procedure Step 3, add after "The clock is now set to 2KHz" this note:

Note: If unable to reach 2KHz, add R1 and R2 as shown
in Figure 25-1.

Page 9: Experiment 25-4 VCO OSCILLATOR SCHEMATIC:

Change "+R" on top of X POT to "1"

Remove FOLLOWER element from schematic: Replace with wire.

Add, Inverter between EXTERNAL SIGNAL GENERATOR and T input
of FLIP-FLOP #1.

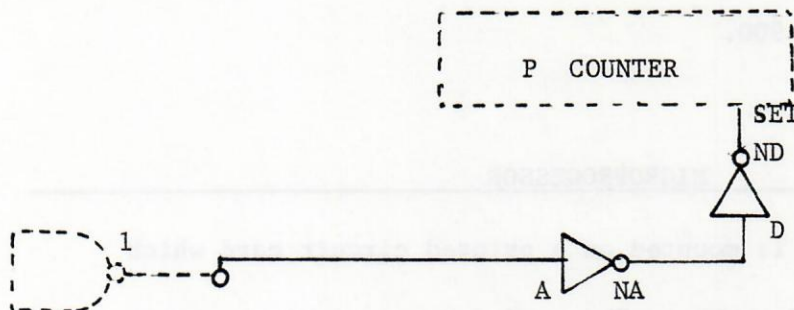
Change  symbol on Speaker to "+P".

ERRATA
ED-LAB 900 ADVANCED DIGITAL SYSTEMS
For Edition Dated 1977

EXP. NO. 25

Page 13: Experiment 25-5 PHASE LOCKED LOOP Schematic

Remove FOLLOWER Symbol in Dotted Box in upper left hand corner of the schematic. Replace with a direct connection. Replace the line from the output of NAND #1 to the SET input of $\div N$ COUNTER with the following two inverters:



Change "+R" at top of X POT symbols with "1"

ADD the following note with arrow to Q of FLIP-FLOP #4.

Note: If the signal generator frequency cannot be adjusted for the lowest frequency measurements, FLIP-FLOP #5 may be inserted at this point to provide an additional division by 2.

Page 15: In the Table, change "FREQ. to FREQ. =
2T x Scale 1
Hz 2T x Scale
Hz

Page 16: PROCEDURE (PLL Closed Loop):

Step 1, line 2 change "+R" to "1"

Page 17: Table Column marked SET, line marked 1800, change "01101 = 14"
to "01110 = 14"

IV. ADD-ONS: ED-LAB 101, ED-LAB 380 and ED-LAB 650

Ed-Lab 101

INTEGRATED CIRCUITS BREADBOARD

The Ed-Lab 101 is an Integrated Circuit Breadboard where the student gains familiarity with the Integrated Circuit Elements. The individual I.C.'s are interconnected on the card which then plugs into the socket on the Ed-Lab 900. The student projects can be connected to and tested on the Ed-Lab 900.

Ed-Lab 380

MICROPROCESSOR

The Ed-Lab 380 is mounted on a printed circuit card which plugs into the Ed-Lab 900 socket. The card contains a Microprocessor I.C. and a Memory I.C. All I.C. connector points are available on the Trainer Panel. The student interconnects the Ed-Lab 700 and 900 logic elements to result in a complete operating Microprocessor System. The Ed-Lab 380 requires a modification of the Ed-Lab 900 which includes the replacement with a Hexadecimal Keyboard and a Hexadecimal Display, and modifications to the Power Supply and other wiring. It will not affect the normal operation of the Ed-Lab 900.

Ed-Lab 650

ELECTRONICS TRAINER

The Ed-Lab 650 is a compatible but separate trainer covering Basic, Intermediate, and Advanced Electronics; including DC, AC, Power Supplies, Transistors, Communications, Pulse Circuits, Industrial Electronics, Controls and other related topics.

V. E D - L A B 9 0 0

ANSWERS TO EXPERIMENTS

V. ED - LAB 900

INTRODUCTION AND COUNTERS

Exp. 1-1 Counter B

1. The count increases by 1 each time switch P is depressed. The COUNT B Input operates properly.

The count ranges from 000 to 111 or from 0 to 7. COUNTER B counts properly.

2. With Switch K DOWN, then UP, counter output clears to 000. The CLEAR Input operates properly.

3. and 4. Set Switches to:

G	H	J
1	1	0

✓ just a typical input

When F is DOWN, then UP, Lamps 6, 7, 8, Display 110. The S0, S1 and S2 Inputs and the SET input operates properly.

INSTRUCTOR NOTES:

1. SWITCHES F & K are the SET and CLEAR to the counter. These must both be UP or "1" to disable these functions. The SET and CLEAR are activated by a "0" input.

SWITCH P provides a single pulse for each push. (i.e. each change of 1 to 0)

The COUNTER B is a BINARY COUNTER, each time the P switch is pushed it will advance to the next count in the binary sequence. Note that Lamp 6 is the most significant digit and Lamp 8 is the least significant digit. The 3 output bits are Lamps 6,7,8.

2. Operating switch K DOWN, then UP, momentarily GROUNDS the counter CLEAR and CLEARS the counter outputs to "000".

3. Switches G, H and J are the PARALLEL INPUTS to the counter and are set to any arbitrary binary number. This number will not appear at the OUTPUT LAMPS until Switch F SET is operated DOWN.

4. Setting SWITCH F DOWN, then UP, momentarily GROUNDS the SET input to the counter. This will SET the binary number from switches G, H, and J into the counter and the number will appear on the OUTPUT LAMPS, 6, 7, and 8. It is not possible to SET the counter without first using the CLEAR.

INTRODUCTION AND COUNTERS

Exp. 1-2 Preset Counter "P"

The Preset inputs S3, S4, S5, S6, and S7 and the SET and CLEAR inputs operate properly.

S7	S6	S5	S4	S3	CHECK
1	0	0	1	0	10010
0	1	1	0	1	01101
0	0	1	1	0	00110

NOTE: These numbers are arbitrary and will differ from student to student.

Exp. 1-3 Cascaded Counters

The counter output display increases by 1 for every four input pulses (Depress P)

QUESTION 1

Counter P is a 5 bit counter so the maximum number of different counts is 32.

QUESTION 2

Counter B is a 3 bit counter so the maximum number of states is 8.

QUESTION 3

Counter B followed by counter P is an 8 bit counter so the maximum number of different counts is 2^8 or 256, with binary numbers ranging from 0 to 255.

QUESTION 4

The highest count for counter P (5 bits) is 11111 = $1 + 2 + 4 + 8 + 16$ resulting in a highest count of 31.

INTEGRATION AND COUNTING

Ex. 1-1 Binary Counter "A"

The counter inputs $2S$, $2R$, $2C$, and $2T$ and the SET and CLEAR inputs operate properly.

2T	2C	2R	2S	2C	2R
1	0	0	1	0	0010
0	1	1	0	1	0101
0	0	1	1	0	0010

NOTE: These numbers are arbitrary and will differ from student to student.

Ex. 1-2 Counter "B"

The counter output display functions by 1 for every four input pulses (count 4).

QUESTION 1

Counter 1 is a 3-bit counter so the maximum number of different counts is 8.

QUESTION 2

Counter 2 is a 3-bit counter so the maximum number of states is 8.

QUESTION 3

Counter 3 follows by counter 2 is an 8-bit counter so the maximum number of different counts is 256, with binary numbers ranging from 0 to 255.

QUESTION 4

The display count for counter 3 (3 bits) is 1111 = 1 + 2 + 4 + 8 = 15 counting to a highest count of 15.

DISPLAYS REGISTER AND OCTAL CODINGExp. 2 - 1 Seven Segment Display

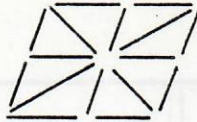
Paragraph 4:

NUMBERS	Display	LETTERS	Display
0		A	
1		C	
2		E	
3		F	
4		H	
5		J	
6		L	
7		P	
8		U	
9		= (equal)	
		- (minus)	
$\frac{1}{2}$		a	
.3		e	
		d	

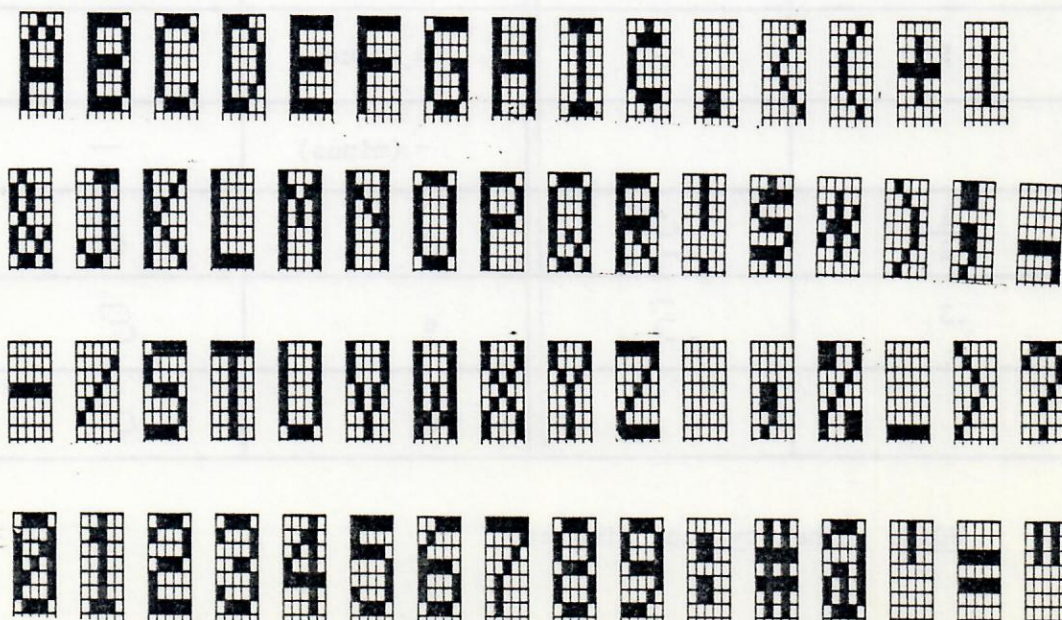
NOTE: Other typical displays.

DISPLAYS REGISTER AND OCTAL CODINGQUESTION 1 & 2

Some early forms of line segments were in the shape of a "star" as shown below using 16 segments:



Most commonly used alphanumeric displays produce all the letters, numbers and punctuation marks using a dot pattern. One is a 5 x 7 matrix of dots and the other is a 7 x 10 matrix of dots. Each dot or square in the matrix can be illuminated so that all letters, numbers and punctuations can be formed. Examples are shown below:

7 x 10 MATRIX5 x 7 MATRIX

DISPLAYS REGISTER AND OCTAL CODING

Exp. 2-2 Binary Coded Decimal

TABLE A

G	H	J	K	SYMBOL
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	BLANK

THESE SYMBOLS HAVE NO
USE IN BCD CODE.

← BLANK can be used in BCD.
When most significant digit
is zero we usually omit it,
or blank it. Thus instead
of 07, we just write 7.

Exp. 2-3 Shift Register

Display Lamps 6 7 8 9 10

1	1	0	1	0
---	---	---	---	---

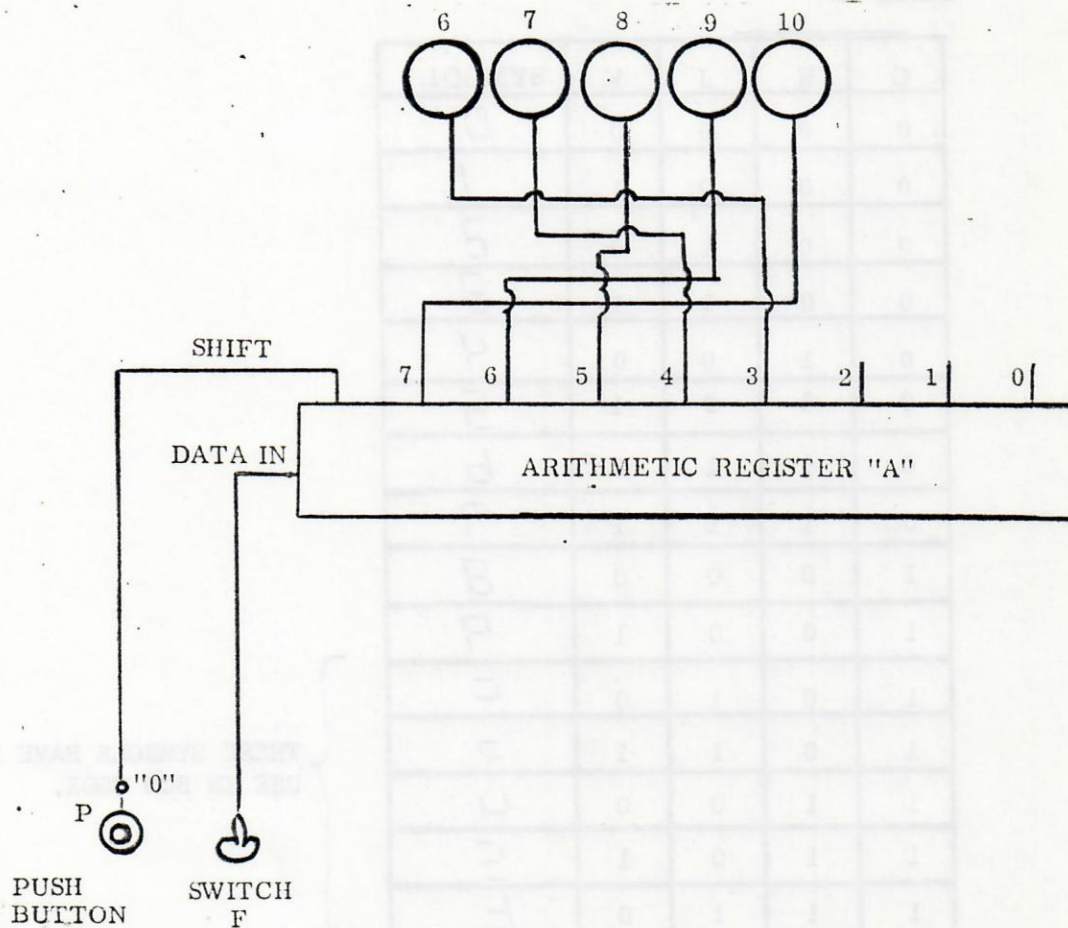
NOTE

The number is entered into the shift register least significant bit first. Thus the order that the bits are entered is 01011. After each bit is entered or set using Switch F it is shifted to the right by depressing Push Button P on the Ed-Lab 900. Therefore the first bit entered ends up all the way to the right in the least significant bit position.

DISPLAYS REGISTER AND OCTAL CODING

QUESTION 3

Diagram to make shift register appear to shift to the left.



Certain types of shift register integrated circuits have provisions for both shift left and shift right.

DISPLAYS REGISTER AND OCTAL CODING

EXP 2-4 BCD Register Display

TABLE B

TEST	DATA IN BINARY	Display
1	00110001 FIRST ↓	31
2	01100111	67
3	10010101	95
4	11000101	45
5	10001010	8C
6	01100010	62
7	01101111	6

← NOT VALID BCD DIGIT
ON THE LEFT

← NOT VALID BCD DIGIT
ON THE RIGHT

← RIGHT HAND DIGIT BLANK

QUESTION 4

The read out will be 31. The 62 in BCD is 01100010 and when it is shifted to the right one position it will become 00110001 which will read out as 31. Note that a shift right results in a division by 2 in this case. It always results in a division by two with "binary" numbers, and with octal "numbers".

DISPLAYS REGISTER AND OCTAL CODING

Exp. 2-5 Octal Register Display

TABLE C

TEST	OCTAL NO.	BINARY (fill in)	DISPLAY	DECIMAL EQUIV.
1	013	00001011 ^{FIRST}	013	11
2	206	10000110	206	134
3	127	01010111	127	87
4	035	00011101	035	29
5	062	00110010	062	50
6	123	01010011	123	83
7	321	11010001	321	209
10	149	01100???	—	—
		NINE DOES NOT EXIST IN OCTAL		
11	222	10010010	222	146
12	325	11010101	325	213

QUESTION 5

The highest number that can be displayed is 377 in octal which is 255 in decimal. The corresponding binary number is 11111111.

QUESTION 6

The advantage of octal coding over binary coded decimal in a binary system is that octal coding utilizes all states of the binary system. In the binary coded decimal system the binary numbers 10 thru 15 are not used in each digit. Octal is actually normal binary numbers broken into groups of 3 bits per digit. Therefore octal is compatible with any binary system.

DISPLAYS REGISTER AND OCTAL CODING

Exp. 2-6 Octal Fraction Display

TABLE D

TEST	OCTAL NO.	BINARY (fill in)	DISPLAY	DECIMAL EQUIV.
1	.232	$\begin{array}{ccccccc} & 2 & & 3 & & 2 & \\ & \underbrace{} & & \underbrace{} & & \underbrace{} & \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$ missing	.2 3 2	.301
2	.726	11101011	.7 2 6	.918
3	.776	111 111 11	.7 7 6	.996
4	.621*	$\begin{array}{ccccccc} 110 & 010 & 01 \\ 110 & 010 & 00 \end{array}$ or	.6 2 2 .6 2 0	.793 .781
5	.002	000 000 01	.0 0 2	.004

*Note in experiment, round off to nearest number.

Note the positional value system; positions multiplied by 2 to the left, and divided by 2 to the right of the "binary point".

etc. 16 8 4 2 1 1/2 1/4 1/8 1/16 etc.

1 1 1 0 1 . 0 1 1 0

The position values for 8 bits in decimal are:

.1	.5
.01	.25
.001	.125
.0001	.0625
.00001	.03125
.000001	.015625
.0000001	.0078125
.00000001	.00390625

The equivalent decimal number can be determined by summing the proper values from the above chart. Note that for 8 bits, the resolution is approx. .004 and therefore, any answer should only be given to 3 decimal places.

Thus 11000000 = .5 + .25 = .750 and .00001000 = .031

DISPLAYS REGISTER AND OCTAL CODING

The equivalent decimal number can also be determined by calculating the normal positive number (154 in the first example) and then dividing by 256, which corresponds to the binary point shifted 8 places to the left. Thus $11000000 = 64 + 128$

$$\text{Thus } 11000000 = \frac{64 + 128}{256} = \frac{192}{256} = .75$$

QUESTION 7

The maximum number that can be displayed on the octal fractional readout is .776 in octal. The binary numbers for this fraction is .11111111. The equivalent decimal number is slightly under 1. When used in computers, it is usually considered that the fractional number is between 0 and 1.

QUESTION 8

The Resolution of the display is 0.002 in octal or $\frac{1}{256}$ (.004) in decimal

Exp. 2-7 BCD Magnitude/Sign Display

TABLE E

TEST	BCD NO.	BINARY (fill in)	DISPLAY
1	- 14	10010100	-14
2	18	00011000	18
3	39	00111001	39
4	- 79	11111001	-79
5	+ 05	00000101	05

Plus is
assumed when
no sign is
given.



DISPLAYS REGISTER AND OCTAL CODING

QUESTION 9

The maximum number that can be displayed with BCD/Sign format is 79: The reason that 99 cannot be displayed is that the 8th bit is used for the sign.

QUESTION 10

The smallest Resolution is 1. The percent resolution is $\frac{1}{79} \times 100\% = 1.26\%$

QUESTION 11

The BCD with sign format allows the use of negative numbers directly in the display. Both positive and negative numbers can be easily read by an operator.

QUESTION 12

The best register and display format would probably be BCD with sign and decimal point from a human factors point of view. From a machine point of view, the octal is the best because it converts directly to binary. Various computers use BINARY (just lamps) or OCTAL or BCD or HEXADECIMAL Displays. Hexadecimal groups the binary numbers in groups of 4 bits to give 16 different combinations; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. When seven segments are used, it is necessary to distinguish between the 6 and the B. The 6 is given by  and the B is given by  (open top). An example is:

$\begin{array}{cc} \overbrace{0110} & \overbrace{1100} \\ 6 & C \end{array}$

The Binary is 01101100.

The Octal is 154.

The Hexadecimal is 6C.

Exp. 2-8 Segment "a" Logic

THIS EXPERIMENT ONLY APPEARS IN THE NEWEST EDITION DATED 1977.

(The logic elements are connected to the top lamp segment "a" as determined by the 4 input binary switches.)

STEP 2 Yes, segment "a" is ON (for binary numbers 2, 3, 5, 7, 8, 9, 0)

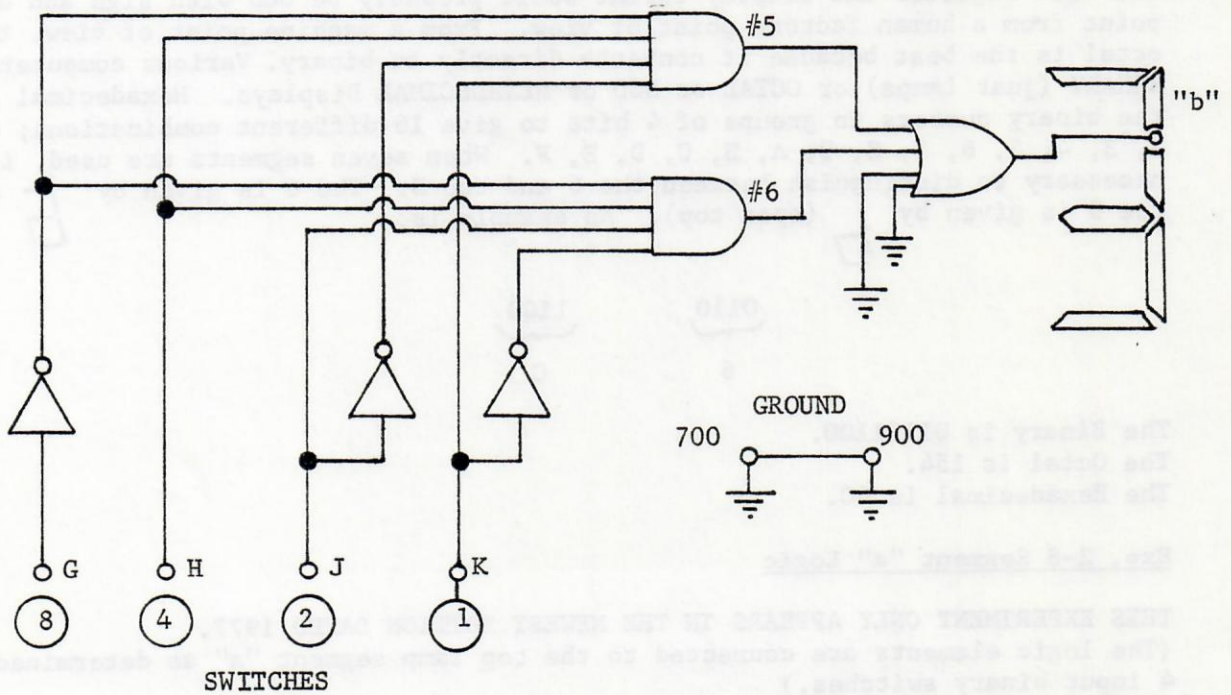
STEP 3 Segment "b" is ON (b = 0) for "1 or 2 or 3 or 4 or 7 or 8 or 9 or 0"
"b" is OFF (b = 1) for "5 or 6"

NOTE: G H J K
5 is 0 1 0 1 (SEE LOGIC DIAGRAM ON NEXT PAGE)
6 is 0 1 1 0

Yes, the logic works. Segment "b" is ON for binary numbers 1, 2, 3, 4, 7, 8, 9 and 0.

DISPLAYS REGISTER AND OCTAL CODING

DIAGRAM A



KEYBOARD DATA INPUT

Exp. 3-1 Keyboard Coding

KEY	8	4	2	1	← BIT CODE
	7	8	9	10	
0	1	0	1	0	← LAMP #
1	1	1	1	0	
2	1	1	0	1	
3	1	1	0	0	
4	1	0	1	1	
5	1	0	1	0	
6	1	0	0	1	
7	1	0	0	0	
8	0	1	1	1	
9	0	1	1	0	
+	0	1	0	0	
-	0	0	1	1	

The STROBE output "S" is a "0" when any key is depressed. The DELAYED STROBE output "DS" is a negative pulse which occurs approximately 5 milliseconds after any key is depressed. The DS pulse is very short. If DS is connected to a Lamp, the fast or short "off" blink is not seen. If DS is connected through an Inverter to a Lamp, the fast "ON" blink will be observed.

QUESTION 1

The result of pushing two keys simultaneously is to connect the codes for both of those keys to ground.

Numbers 1 and 2 may be pushed to obtain the 3 code at the output.

KEYBOARD DATA INPUT

Exp. 3-2 Keyboard to Display

TABLE B

KEY	DISPLAY
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
+	+
-	-

*NOTE: 0, +, - result in binary output codes which are larger than 9. External Logic may be connected to decode these functions.

QUESTION 2

Since Octal Format requires only 3 binary Bits the 8 input to the display would be disconnected from the keyboard and the display 8 input would be connected to ground. For the full octal wiring diagram see diagram experiment logic 3-3.

QUESTION 3

The Highest Digit in octal is a 7.

Exp. 3-3 Data Entry & Octal Display

TABLE C

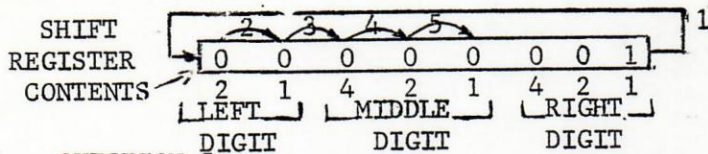
OCTAL	ENTER	DECIMAL
123	01010011	83
172	01111010	122
263	10110011	179
341	11100001	225
256	10101110	174

Selected by student →

KEYBOARD DATA INPUT

QUESTION 4

The pulses from the P Switch shift the register to the right. The bits which are shifted out of the register to the right are fed back into the left end of the shift register. This is called end around shifting. In order to move the 1 in 001 into the center position, so the display will read 010 we must shift 5 times to the right. This process is shown in the diagram below:



SO THE FINAL DISPLAY WILL BE 010

QUESTION 5

The Maximum numeral that can be displayed with the left hand digit is a 3. The 4 and 8 inputs are connected to ground.

Exp. 3-4 BCD Display & Register

BCD DISPLAY	ENTER	DECIMAL NUMBER
99	10011001	99
49	01001001	49
02	00000010	02
26	00100110	26

QUESTION 6

With the number BCD 32 or 00110010 in the register, a single shift will change it to 00011001 or 19 when decoded as a BCD number. Note that when BCD coding is used, the number is not divided in half when shifted to the right.

REGISTER DATA

REGISTER A

The pointer from the 8-bits shifts the register to the right. The bits which are shifted out of the register to the right are fed back into the left end of the shift register. This is called end-around shifting. In order to move the 1 in 001 back to the center position, as the display will read 010 we must shift 3 times to the right. This process is shown in the diagram below.



The maximum number of times that can be displayed with the left hand digit is 3. The 4 and 5 inputs are connected to ground.

Fig. 3-4 BCD Display A Register

BCD DISPLAY	ENTER	DECIMAL NUMBER
00	00010001	00
01	00010010	01
02	00010011	02
03	00010100	03

REGISTER B

When the number 001 00 or 001100 is in the register, a single shift will change it to 0001001 or 10 when decoded as a BCD number. Note that when BCD coding is used, the number is not divided in half when shifted to the right.

RANDOM ACCESS MEMORYExp. 4-1 Random Access, Write-Read

ADDRESS				BIT OF DATA	
G	H	J	K	SET IN	READ OUT
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	0	0

DATA IS ARBITRARILY
SET-IN BY STUDENT

Word Memory

WORD ADDRESS		SET J, K (BIT ADDRESS)		WORD READ - OUT	CHECK
LOCATION G, H	DECIMAL WORD INPUT	BINARY WORD 4 BITS			
0 (00)	5	0101		0101	✓
1 (01)	6	0110		0110	✓
2 (10)	9	1001		1001	✓
3 (11)	3	0011		0011	✓

RANDOM ACCESS MEMORY

Location and numbers chosen by student. They will differ from report to report.

WORD ADDRESS	USE J, K FOR BIT ADDRESS		
LOCATION (G, H)	WORD INPUT	WORD READ - OUT	CHECK
0	2 (0010)	2 (0010)	✓
1	1 (0001)	1 (0001)	✓
2	3 (0011)	3 (0011)	✓
3	5 (0101)	5 (0101)	✓
0	7 (0111)	7 (0111)	✓
1	2 (0010)	2 (0010)	✓
2	6 (0110)	6 (0110)	✓

The word in location number 2 is changed by:

1. Selecting Word 2 with switches G and H.
2. Setting J and K at "0"
3. Setting data to the first bit of the new number
4. Operate Read/Write into "Write" then back to "Read" to set the new bit.
5. Change bit address (J & K) and enter other bits.

QUESTION 1

64-4 bit words can be stored in a 256 bit memory: $\frac{256}{4} = 64$

QUESTION 2

32-8 bit words can be stored in the memory: $\frac{256}{8} = 32$

QUESTION 3

42-6 bit words can be stored in the memory: $\frac{256}{6} = 42\frac{1}{2}$ rounded off to 42

QUESTION 4

The longest word that can be stored in the memory is 256 bits (one word).

RANDOM ACCESS MEMORYQUESTION 5

Using binary-coded-decimal, each 9 digit social security number uses (9×4) 36 bits so 7 social security numbers can be stored in the memory, resulting in 252 bits; 4 remain left over.

Exp. 4-2 Memory and Display

ADDRESS BITS 2,1,0



MEMORY ADDRESS					ENTER BCD DATA Switches H, J, K		READ - OUT
7	6	5	4	3	TENS	ONES	
O	O	O	1 (OPEN)	0	0011	0010	32
O	O	O	1	1	0100	0111	47
O	O	O	0 (GND)	0	1001	1000	98
1	1	1	0	1	1000	0111	87
1	1	1	1	0	0101	0110	56
1	1	1	0	1	0110	0010	62
1	1	0	1	1	0111	1001	79
1	1	0	0	0	0010	0101	25
1	0	1	0	0	0101	0001	51
1	0	1	1	0	1000	0011	83

"0" = grounded input

"1" = opened input

Addresses and data arbitray

QUESTION 6

To store a telephone number in BCD requires 40 Bits (10 numerals, including area code at 4 Bits (BCD) Per number). Using 8 Bit Digital words we would need 5 words to store a complete telephone number with area code.

PHONE NUMBER CODING:

Under word Address, do not use switches H, J, K; Use the Memory Address BITS 6,5,4. Data reads in and out okay.

RANDOM ACCESS MEMORY

Exp. 4-3 Automatic Address

PROCEDURE 5. (page 11) The correct data must be read on the display?

ANSWER: YES

PROCEDURE 6. (page 11) The WORD ADDRESS should increase by 1 each time 8 bits of data are entered or read from the system?

ANSWER: THE ADDRESS IS CORRECT

ADDRESS		DATA	
Binary	Octal	Binary BCD	Decimal
00000	00	01100111	67
00001	01	00110010	32
00010	02	01000001	41
00011	03	01011001	59
00100	04	00111001	39
00101	05	10011001	99
01011	13	01001000	48

THIS DATA IS
SELECTED BY THE
STUDENT

REGISTER PRESETS AND DECODING

Exp. 5-1 Register and Decoder

TABLE A

SWITCHES			DISPLAY			DECODER OUTPUTS							
H	J	K	8 C	9 B	10 A	N7	N6	N5	N4	N3	N2	N1	N0
0	0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	1	0	0	1	1	1	1	1	1	1	0	1
0	1	0	0	1	0	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1	1

With Switches F & G both DOWN the output lamps follow the INPUT SWITCHES H, J, K.

With EN not connected to ground all the DECODER OUTPUTS are always 1. The decoder is disabled. (i.e. all outputs are 1)

QUESTION 1

There are 16 outputs for a 4 input decoder. $2^4 = 16$.

LOGIC CIRCUITS AND LOGIC

Fig. 1-1: Decoder and Display

TABLE 1

SWITCHES	DISPLAY		DECODER OUTPUTS									
	H	L	A	B	C	D	E	F	G	H	I	J
0 0 0	0	0	0	0	0	0	0	0	0	0	0	0
0 0 1	0	0	0	0	0	0	0	0	0	0	0	0
0 1 0	0	0	0	0	0	0	0	0	0	0	0	0
0 1 1	0	0	0	0	0	0	0	0	0	0	0	0
1 0 0	1	0	0	0	0	0	0	0	0	0	0	0
1 0 1	1	0	0	0	0	0	0	0	0	0	0	0
1 1 0	1	0	0	0	0	0	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0	0	0	0	0

With switches T & S both down the output lamps follow the input switches H, I, J, K. With H not connected to ground all the DECODER OUTPUTS are always 1. The decoder is disabled (i.e., all outputs are 1).

QUESTION 1

There are 10 outputs for a 4 input decoder. $2^4 = 16$.

MULTIPLEXING AND DATA TRANSFERExp. 6-1 Preset Instruction RegisterTable A

SWITCH - INPUTS					LAMP - OUTPUTS				
F	G	H	J	K	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	1	0
1	1	1	0	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	0	0	1
0	0	1	1	0	0	0	1	1	0
1	0	0	1	1	1	0	0	1	1

These
numbers are
chosen by
the student

MULTIPLEXING AND DATA TRANSFER

Exp. 6-2 C Preset-Data Transfer

TABLE B

OPERATION	SWITCHES					LAMP OUTPUTS				
	F	G	H	J	K	6	7	8	9	10
PRESET "P"	1	0	0	1	1	1	0	0	1	1
COUNT P	1	0	0	1	1	1	0	1	0	0
COUNT P	1	0	0	1	1	1	0	1	0	1
COUNT P	1	0	0	1	1	1	0	1	1	0
PRESET "P"	0	1	1	1	0	0	1	1	1	0
COUNT P	0	1	1	1	0	0	1	1	1	1
COUNT P	0	1	1	1	0	1	0	0	0	0
COUNT P	0	1	1	1	0	1	0	0	0	1
COUNT P	0	1	1	1	0	1	0	0	1	0
COUNT P	0	1	1	1	0	1	0	0	1	1

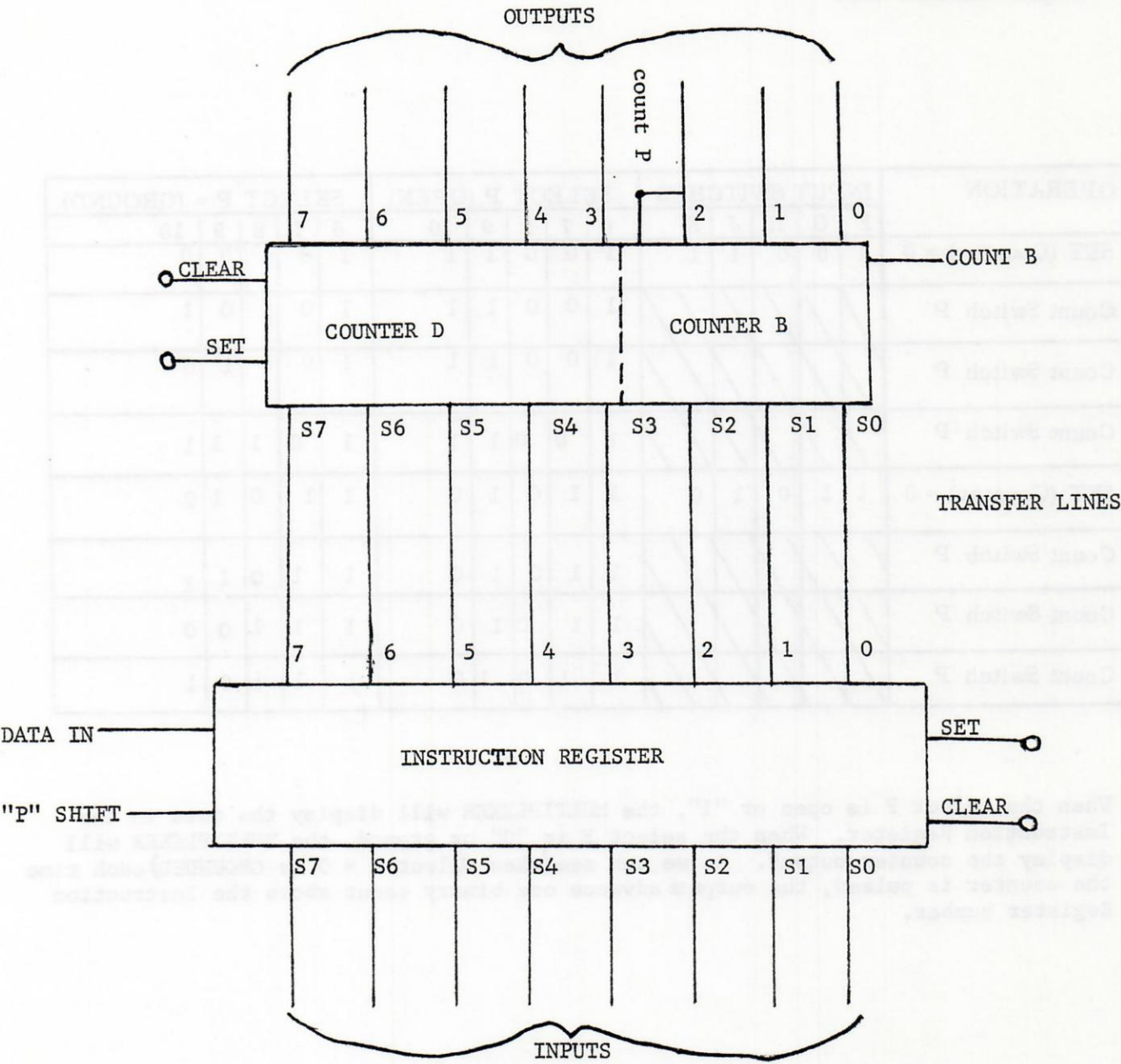
QUESTION 1

The program counter will not count when the set line is connected to ground because the set line will force the outputs to agree with the set inputs.

QUESTION 2

Circuit for 8 Bit Program counter with 8 Bit instruction register and 8 Bits of transfer.

MULTIPLEXING AND DATA TRANSFER



MULTIPLEXING AND DATA TRANSFERExp. 6-3 Multiplexer

OPERATION	INPUT SWITCHES					SELECT P (OPEN)					SELECT P = (GROUND)				
	F	G	H	J	K	6	7	8	9	10	6	7	8	9	10
SET (Counter) = 0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	0
Count Switch P	/	/	/	/	/	1	0	0	1	1	1	0	1	0	1
Count Switch P	/	/	/	/	/	1	0	0	1	1	1	0	1	1	0
Count Switch P	/	/	/	/	/	1	0	0	1	1	1	0	1	1	1
SET (Counter) = 0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0
Count Switch P	/	/	/	/	/	1	1	0	1	0	1	1	0	1	1
Count Switch P	/	/	/	/	/	1	1	0	1	0	1	1	1	0	0
Count Switch P	/	/	/	/	/	1	1	0	1	0	1	1	1	0	1

When the select P is open or "1", the MULTIPLEXER will display the data in the Instruction Register. When the select P is "0" or ground, the MULTIPLEXER will display the counter output. As we can see (when Select P = 0 is GROUNDED) each time the counter is pulsed, the outputs advance one binary count above the Instruction Register number.

ADDER AND OCTAL ARITHMETICExp. 7-1 Full AdderTRUTH TABLE A

A	B	C	SUM (S)	CARRY (K)	NOT SUM (N S)
0	0	0	0	0	1
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	0

THE SUM AND CARRY RESULTS ARE CORRECT.

ADD THESE NUMBERS:

A 11001010
 + B 00010111
 SUM 11100001

TABLE B

LEAST SIGNIFICANT BIT FIRST

	128	64	32	16	8	4	2	1	DECIMAL
A	1	1	0	0	1	0	1	0	202
+ B	0	0	0	1	0	1	1	1	23
C	0	0	1	1	1	1	0	0	X
(K) CARRY	0	0	0	1	1	1	1	0	
(S) SUM	1	1	1	0	0	0	0	1	
									225

EB = 0 SET C = 0

When EB = 0, the B input to the adder is disabled and appears to be "0" regardless of the input switch condition. EB = 0 disables the B input to the adder.

A	B	S
0	0	0
0	1	0
1	0	1
1	1	1

ADDER AND OCTAL ARITHMETIC

EA = 0 also Switch C = 0

A	B	S
0	0	0
0	1	1
1	0	0
1	1	1

EA=0 sets the A input to the adder at "0" regardless of the input switch condition. EA=0 disables the A input to the adder.

Exp. 7-2 Adder with Memory

TABLE E

least significant
enter first
↓

DECIMAL
NUMBER

OCTAL
NUMBER

A	0	0	1	1	1	1	1	1	63	077
+ B	0	1	0	1	1	1	0	1	93	135
(SUM) S	1	0	0	1	1	1	0	0	156	234

A	0	1	0	0	1	1	0	1	77	115
+ B	0	0	1	0	0	1	0	0	36	044
S	0	1	1	1	0	0	0	1	113	161

A	0	0	0	0	0	1	1	1	7	007
+ B	0	0	1	1	0	1	1	0	54	066
S	0	0	1	1	1	1	0	1	61	075

ADDER AND OCTAL ARITHMETICQUESTION 1

Convert Binary Numbers to Octal:

- (a) 0 0 0 1 0 1 1 1 BINARY --- 027 OCTAL
 (b) 1 1 0 1 0 0 1 0 BINARY --- 322 OCTAL
 (c) 1 1 1 0 1 0 0 1 BINARY --- 351 OCTAL

QUESTION 2

When using Octal or Binary numbers with sign, the left hand bit represents the sign of the number. When the left hand bit is a 1, the sign is negative and the number is in 2's complement or 1's complement +1. To obtain the magnitude, reverse bits or 1's complement and add 1; left hand bit is a 0 the sign is positive.

- (a) 0 0 0 1 0 1 1 1 = +23 (decimal)
 (b) 1 1 0 1 0 0 1 0 = (0 0 1 0 1 1 0 1 + 1) = 0 0 1 0 1 1 1 0 = -46 (decimal)
 (c) 1 1 1 0 1 0 0 1 = (0 0 0 1 0 1 1 0 + 1) = 0 0 0 1 0 1 1 1 = -27 (decimal)

QUESTION 3

- (a) This number is a 3 because each word is only 8 Bits Long.

$$\begin{array}{r} 378 \\ - 024 \\ \hline \text{ANS. } 354 \end{array}$$

$$\begin{array}{r} (b) \quad 378 \\ - 034 \\ \hline \text{ANS. } 344 \end{array}$$

$$\begin{array}{r} (c) \quad 378 \\ - 257 \\ \hline \text{ANS. } 121 \end{array}$$

$$\begin{array}{r} (d) \quad 378 \\ - 347 \\ \hline \text{ANS. } 031 \end{array}$$

NOTE: In each case, the answer plus the number to be complemented, add to zero. The 8's complement is important, because negative answers in the computers appear in octal as 8's complement.

EXAMPLE:

024	number
+ 354	8's complement
<u>110</u>	Carry
400	SUM
000	Because the left hand digit has only 2 bits, the 4 will not exist and the digit will read "0".

QUESTION 4

(a)	Octal		Decimal
	035		29
	+ 027	DECIMAL	+ 23
	064 = $6 \times 8 + 4 = 48 + 4 = 52$		52

(b)	125		85
	+ 032	DECIMAL	+ 26
	157 = $(64 \times 1) + 5(8) + 7 = 64 + 40 + 7 = 111$		111

(c)	077		63
	+ 136	DECIMAL	+ 94
	235 = $2(64) + 3(8) + 5 = 128 + 24 + 5 = 157$		157

235 is not a valid positive number because the "1" in most significant digit means a negative number.

ADDER AND OCTAL ARITHMETIC

Octal

Decimal

(d) 077
+236
335 = 3 (64) + 3 (8) + 5 = 192 + 24 + 5 + 221

63
+158
221

{ SAME COMMENT AS ABOVE.

(e) 372
+157
551 = 5 (64) + 5 (8) + 1 = 320 + 40 + 1 = 36

250
+111
361

{ ALTHOUGH NUMBERS CHECK, 551 is not a valid number. With 8 bits, the left digit could only reach 3.

(f) 072 ---COMPL 072
-157 +221 (on the 8 bit machine)
(subtract 331 (on the 8 bit machine)
from 378)

58
-111
-53

313 COMPL. = -065 Octal = -53 decimal

(g) 157 153
-072 ----COMPL----+306
(overflow) 065

111
-58
53

The left hand bit (4 Bit) is lost when working with an 8 bit word.
The Octal answer is wrong because of the "OVERFLOW".

(h) 234 ----COMPL--- 234 → Note: (234 is negative; complement is 144) → -100
-321 +057 (display) or -100 decimal → -047
313
-065 Octal (321 is negative; complement is 057) → -100
= -53 decimal or -47 decimal → -047
-53

(i) 234 ----- 234
-121 COMPL. +257 (234 is negative; complement is 144) → -100
113 (positive number) or -100 decimal → -81
(Overflow 10) = +75 decimal (121 is +81 decimal) → -181 decimal
(incorrect)

NOTE: Proper answer is (-181) decimal. The complement system results in an answer of (+75) decimal, which is an error. This is because there is a "negative overflow". The maximum negative number in the 8 bit complement notation is (200)(Octal) which corresponds to (178) octal or (-128) decimal. Therefore, -181 decimal cannot be represented in the 8 bit negative notation.

ADDER AND OCTAL ARITHMETIC

Exp. 7-4 & 7-5 Bit Delay Master Slave & Triggering

COMMENT: In one sentence "Do not change the J & K inputs when T is at a "1".

Exp. 7-6 Serial Register and Adder

4.

		ENTER ↓	READ ↓	
	NUMBER	BINARY	SUM	CHECK
	035	00011101	035	035
	027	00010111	064	064
RESET "A" (Switch J) →	125	01010101	125	125
	032	00011010	157	157
RESET "A" (Switch J) →	077	00111111	077	077
	236	10011110	335	335

CHECKS

5.

NUMBER	BINARY (Enter)	SUM	CHECK
035	00011101	35	035
027	00010111	64	064
125	01010101	211	211
032	00011010	243	243
077	00111111	342	342
035	00011101	377	377
236	10011110	(OVERFLOW)	235

ADDER AND OCTAL ARITHMETICExp. 7-7 Instruction Arithmetic Unit

COMMAND			DATA "B"		A REGISTER	CHECK
	Octal	Switch GHJ Binary	Binary	Octal	DISPLAY	
LOAD	6	1 1 0	00000100	004	004	004
LOAD	6	1 1 0	00000111	007	007	007
ADD	5	1 0 1	00100011	043	052	052
ADD	5	1 0 1	01010011	123	175	175
SUBTRACT	4	1 0 0	00100100	044	131	131
STORE	2	0 1 0	11010100	324 *	131	131

4. Yes. The register outputs agree with the expected results.

5.

SET-IN ↘	ENTER ↘		
COMMAND: Set Switches G, H, J	DATA "B": Set Switch F (Serially)	DISPLAY A Register	CHECK
LOAD (W) 110	074	074	074
ADD (X) 101	116	212	212
SUB (Y) 100	024	166	166
SUB (Z) 100	031	135 (ANSWER)	135

6. Yes. The outputs agree with expected results.

Exp. 7-8 Programmed Memory & Arithmetic Unit

4. Yes. All data words are properly displayed.

5. Yes. The display shows the correct results

ADDER AND OCTAL ARITHMETIC

Exp. 7-9 Line Frequency Counter

	OCTAL READING	CALCULATE TIME
CLEAR	000	0 SECONDS
START-STOP	060	0.80
START-STOP	122	1.36
START-STOP	275	3.15

} numbers will vary with student tests.

Typical Calculation

060 (octal) = 48 decimal

Time = 48 counts $\frac{1}{60}$ second = $\frac{48}{60}$ seconds

Time = 0.80 seconds

AUTOMATIC DIGITAL COMPUTERINSTRUCTORS NOTE:

This experiment outlines the architecture of the Automatic Digital Computer and details the function of each section of the Computer Schematic. Also the Timing sequence of the operations of the computer and the waveforms expected at key points in the schematics are given.

The timing logic is wired and the start button is operated. The student observes the computer going from OFF to ON (and back to OFF in ONE STEP). He also observes the FETCH/EXECUTE cycles on Lamps 3 and 4. He also tests the AUTO/ONE STEP operation and the STOP operation in AUTO mode.

This experiment only constructs the timing logic and the computer is completed in Experiment No. 9.

The student could write a test sheet verifying operation. The test sheet could appear as shown below:

PROCEDURE	ITEM	CHECK
3.	OFF; INST LAMPS Lit	✓
4.	ONE CYCLE	✓
7.	AUTO/RUN	✓
8.	AUTO/HALT	✓
11.	TIMING SIGNALS OPTIONAL	✓

AUTOMATIC DIGITAL COMPUTER

INSTRUCTIONS:

This experiment outlines the architecture of the Automatic Digital Computer and details the function of each section of the computer schematic. Also the timing sequence of the operations of the computer and the waveforms reported at key points in the schematic are given.

The timing logic is wired and the action button is connected. The student observes the computer going from OFF to ON (and back to OFF in ONE STEP). He also observes the ERROR/RESTART signal on lamps 3 and 4. He also tests the AUTOMATIC STOP operation and the STOP operation as AUTO mode.

This experiment only constructs the timing logic and the computer is completed in Experiment No. 3.

The student could write a test sheet verifying operation. The test sheet could report as shown below:

EXPERIMENT	ITEM	CHECK
1.	OFF: INIT LAMP 1-4	✓
2.	ONE CYCLE	✓
3.	AUTOMATIC	✓
4.	AUTOMATIC	✓
5.	TIME SIGNAL OFFICIAL	✓

ARITHMETIC AND MEMORY SYSTEM

INSTRUCTORS NOTE:

This experiment completes the construction and check out of the Automatic Digital Computer. Individual Test Programs are run to test each instruction and a Test Maintenance Routine for checking the whole computer operation is completed. The full explanation of the Instructions and the Computer operation is completed.

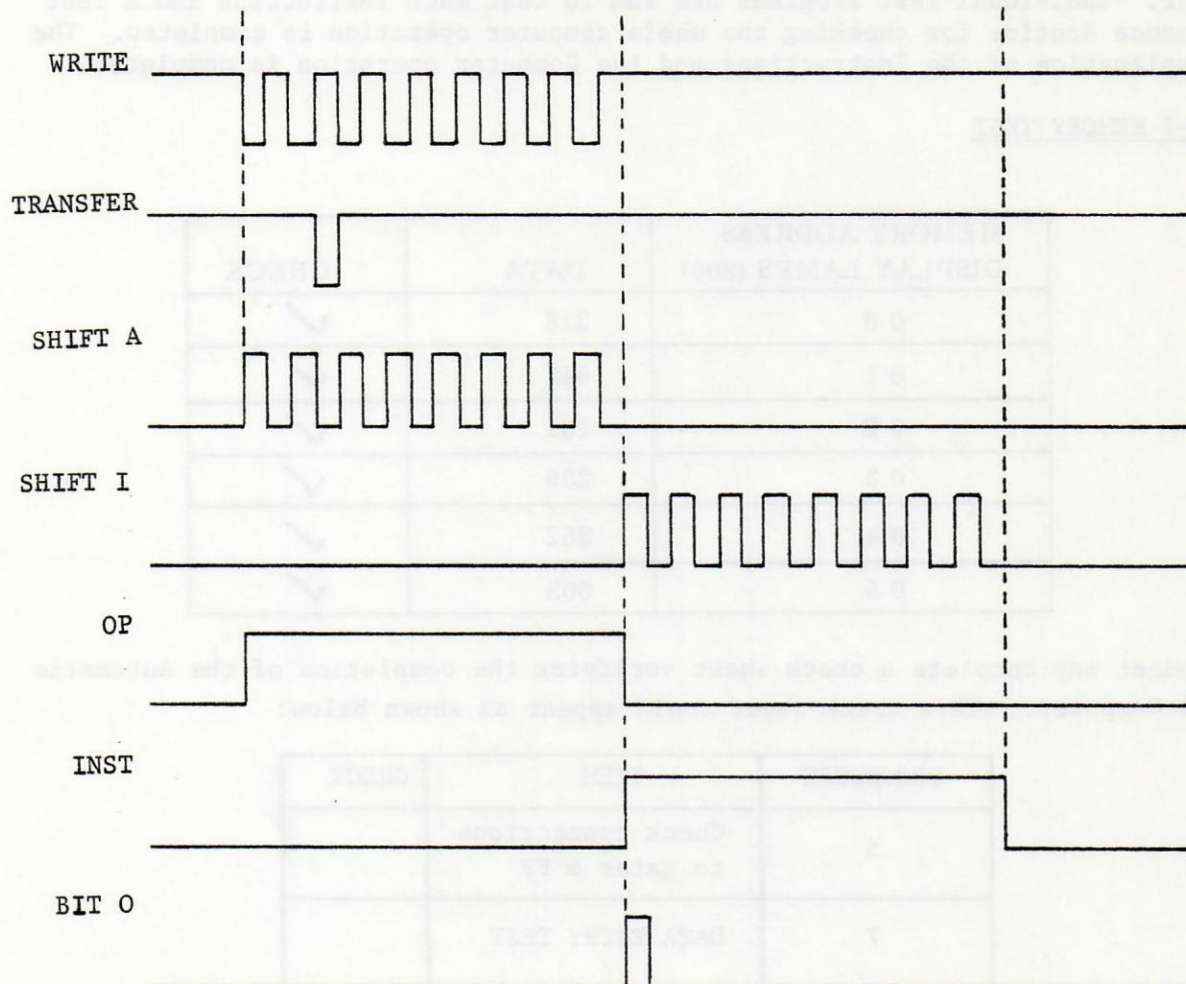
Exp. 9-7 MEMORY TEST

MEMORY ADDRESS DISPLAY LAMPS (900)	DATA	CHECK
0 0	216	✓
0 1	047	✓
0 2	151	✓
0 3	236	✓
0 4	362	✓
0 5	003	✓

The student may complete a check sheet verifying the completion of the Automatic Digital Computer. This check sheet could appear as shown below:

PROCEDURE	ITEM	CHECK
5	Check connections to gates & FF	
7	DATA ENTRY TEST	
8	P COUNTER TEST	
9	MEMORY TEST	

Exp. 9-10.6 Control Pulse Tests



ARITHMETIC AND MEMORY SYSTEM

The following forms may be used by the student to verify the Test Maintenance Routines given in Experiment 9.

1. LOAD TEST (6)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

2. ADD TEST (5)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

3. SUBTRACT TEST (4)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

ARITHMETIC AND MEMORY SYSTEM

4. STORE TEST (2)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

5. SHIFT RIGHT (3)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

6. TRANSFER (1)

PROGRAM READ OUT

PROGRAM RUN

MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

ARITHMETIC AND MEMORY SYSTEM

7. TRANSFER ON NEGATIVE (0)

PROGRAM READ OUT		PROGRAM RUN
MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		
06		
07		

8. HALT (7)

PROGRAM READ OUT		PROGRAM RUN
MEMORY ADDRESS DISPLAY LAMPS (900)	WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		

ARITHMETIC AND MEMORY SYSTEM

12. MAINTENANCE TEST PROGRAM

MEMORY ADDRESS DISPLAY LAMPS (900)	PROGRAM READ OUT WORD READ ON DISPLAY SWITCH F-DOWN, GHJ-110	PROGRAM RUN ACCUMULATOR READ ON DISPLAY SWITCH F-UP
00		
01		
02		
03		
04		
05		
06		
07		
10		
11		
12		
13		
14		
15		
16		
17		
20		
21		
22		
23		
24		
25		

PROGRAMMING AND FLOW CHARTS

PROCEDURE:

PROGRAM TEST $M = F(A, B)$

PROBLEM	A	B	FUNCTION	COMPUTER ANSWER	CHECK
1	016	034	$A + B$	052	✓
2	016	034	$A - B$	362	✓
3	016	034	$B - A$	016	✓
4	016	034	$2A - B$	000	✓
5	021	370	$2A - B$	052	✓
6	370 ¹	362 ²	$A + B$	352 ⁴	✓
7	367 ³	021	$B - A$	032	✓

NOTES: 1) 370 = -010 3) 367 = -011

2) 362 = -016 4) 352 = -026

CHECK OF RESULTS (IN OCTAL)

PROBLEM 1

$$\begin{aligned}
 A &= 016 \\
 B &= 034 \\
 M &= A + B = 016 + 034
 \end{aligned}$$

$$\begin{array}{r}
 016 \\
 + 034 \\
 \hline
 010 \text{ CARRY} \\
 052 \text{ ANSWER}
 \end{array}$$

PROBLEM 2

$$\begin{aligned}
 A &= 016 \\
 B &= 034 \\
 M &= A - B = 016 - 034 = 016 + 344
 \end{aligned}$$

$$\begin{array}{r}
 016 \\
 - 344 \\
 \hline
 010 \text{ CARRY} \\
 362 \text{ ANSWER OR (COMPLEMENT) - 016}
 \end{array}$$

PROBLEM 3

$$\begin{aligned}
 A &= 016 \\
 B &= 034 \\
 M &= B - A = 034 - 016 = 034 + 362
 \end{aligned}$$

$$\begin{array}{r}
 034 \\
 - 362 \\
 \hline
 00 \text{ CARRY} \\
 016 \text{ ANSWER}
 \end{array}$$

PROBLEM 4

$$\begin{aligned}
 A &= 016 \\
 B &= 034 \\
 M &= 2A - B = A(016) - 034 = 034 - 034 = 034 - 344
 \end{aligned}$$

$$\begin{array}{r}
 034 \\
 - 344 \\
 \hline
 110 \text{ CARRY} \\
 000 \text{ ANSWER}
 \end{array}$$

(NOTE the left hand 1 is not displayed)

PROGRAMMING AND FLOW CHARTS

PROBLEM 5

$$\begin{array}{rcl} A & = & 021 \\ B & = & 370 \\ M & = & 2A - B = 2(021) - 370 = 042 + 010 \end{array} \quad \begin{array}{r} 042 \\ - 010 \\ \hline 000 \text{ CARRY} \\ 052 \text{ ANSWER} \end{array}$$

PROBLEM 6

$$\begin{array}{rcl} A & = & 370 \\ B & = & 362 \\ M & = & A + B = 370 + 362 \end{array} \quad \begin{array}{r} 370 \\ + 362 \\ \hline 00 \text{ CARRY} \\ 352 \text{ ANSWER} \end{array}$$

(NOTE that the left hand 1 is not displayed)

352 is -026 ANSWER

CHECK

$$\begin{array}{rcl} A & = & 370 = -010 \\ B & = & 362 = -016 \\ \text{SUM} & = & -026 \end{array}$$

PROBLEM 7

$$\begin{array}{rcl} A & = & 367 \\ B & = & 021 \\ M & = & B - A = 021 - 367 = 021 + 011 \end{array} \quad \begin{array}{r} 021 \\ - 011 \\ \hline 0 \text{ CARRY} \\ 032 \text{ ANSWER} \end{array}$$

CODING CHART FOR X = |A| + |B|

NOTE FOR INSTRUCTOR

Note that Step 04 is never used in the program. The programmer has apparently included this instruction but never used it. Having this extra instruction in the program does not effect the operation of the program.

PROGRAMMING AND FLOW CHARTS

PROGRAM TEST $X = A + B $				
PROBLEM	A	B	X	CHECKED
1	005	002	007	007
2	-005 (373)*	-002 (376)*	007	007
3	010	-012 (366)*	022	022
4	-020 (360)*	010	030	030

* ENTER NEGATIVE NUMBER (2's COMPLEMENT)

PROBLEM 1 (Check):

$$\begin{array}{l} A = 005 \\ B = 002 \end{array} \quad \begin{array}{l} |A| = 005 \\ |B| = 002 \end{array} \quad X = |A| + |B| = 005 + 002 = 007 \text{ ANSWER}$$

PROBLEM 2 (Check):

$$\begin{array}{l} A = 373 = -005 \\ B = 376 = -002 \end{array} \quad \begin{array}{l} |A| = 005 \\ |B| = 002 \end{array} \quad X = |A| + |B| = 005 + 002 = 007 \text{ ANSWER}$$

PROBLEM 3 (Check):

$$\begin{array}{l} A = 010 \\ B = 366 = -012 \end{array} \quad \begin{array}{l} |A| = 010 \\ |B| = 012 \end{array} \quad X = |A| + |B| = 010 + 012 = 022 \text{ ANSWER}$$

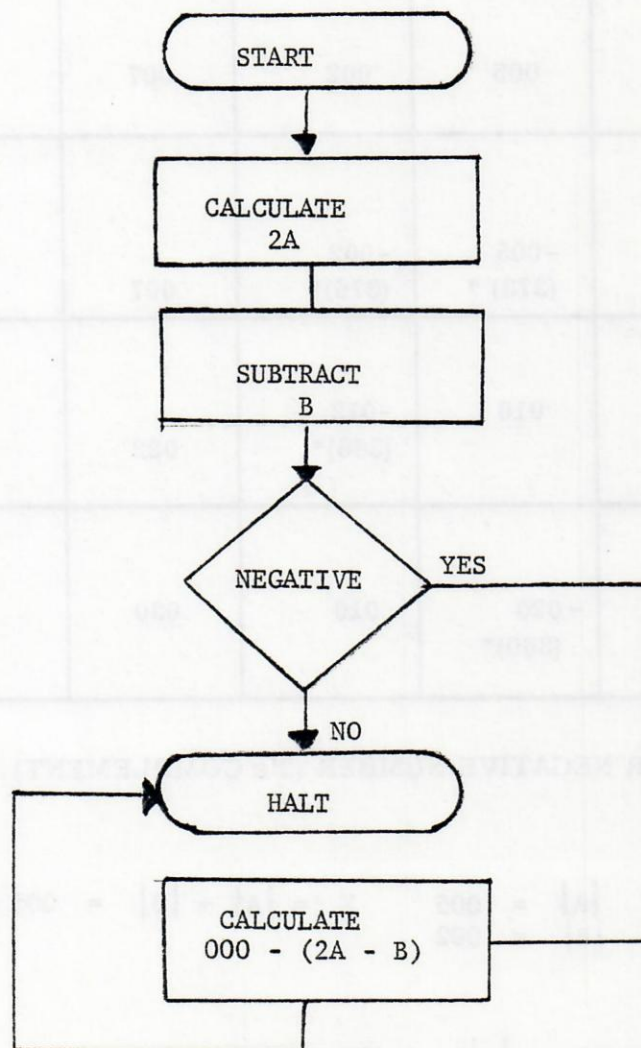
PROBLEM 4 (Check):

$$\begin{array}{l} A = 360 = -020 \\ B = 010 \end{array} \quad \begin{array}{l} |A| = 020 \\ |B| = 010 \end{array} \quad X = |A| + |B| = 020 + 010 = 030 \text{ ANSWER}$$

PROGRAMMING AND FLOW CHARTS

QUESTION 1(a)

FLOW CHART $Y = |2A - B|$



PROGRAMMING AND FLOW CHARTS

1(b) SYMBOLIC PROGRAM $Y = |2A - B|$
 or ASSEMBLY PROGRAM

REFERENCE ADDRESS	WORD	COMMENT
	L(1) TRA	TRANSFER to L(1)
L(A)	$\emptyset$	A
L(B)	$\emptyset$	B
L(1)	L(A) LOAD	LOAD A
	L(A) ADD	CALCULATE 2A
	L(B) SUB	SUBTRACT B
	L(Y) STORE	STORE IN L(Y)
	L(2) TON	TRANSFER ON NEGATIVE TO L(2)
L(3)	L(NO) HALT	HALT (DISPLAY Y)
L(2)	L(NO) LOAD	LOAD 000
	L(Y) SUB	SUBTRACT Y
	L(Y) STORE	STORE ANSWER IN L(Y)
	L(3) TRA	TRANSFER TO L(3)
L(Y)	$\emptyset$	ANSWER
L(NO)	000	ZERO

PROGRAMMING AND FLOW CHARTS

1(c) CODING CHART $Y = 2A - B$

ADDRESS	WORD	COMMENTS
00	02(1)	TRANSFER TO 03
01	$\emptyset$	A
02	$\emptyset$	B
03	01(6)	LOAD A
04	01(5)	ADD A (2A)
05	02(4)	SUBTRACT B
06	15(2)	STORE IN 15
07	10(0)	TRANSFER ON NEGATIVE 11
10	16(7)	HALT
11	16(6)	LOAD ZERO
12	15(4)	SUBTRACT (2A - B)
13	15(2)	STORE IN 15
14	07(1)	TRANSFER TO 10
15	$\emptyset$	ANSWER
16	000	ZERO

1(d) OPERATOR INSTRUCTIONS

STORE A in location 01
STORE B in location 02
START PROGRAM at 00
AUTOMATIC HALT
ANSWER IN DISPLAY AND LOCATION 15

PROGRAMMING AND FLOW CHARTS

1 (e). PROGRAM TEST SHEET

TEST	A	B	(2A - B)	CHECK
1	002	001	003	✓
2	001	010	006	✓
3	374	004	014	✓
4	002	004	000	✓

CHECK (IN OCTAL)

TEST 1	$ 2(002) - 001 = 004 - 001 = 003$
TEST 2	$ 2(001) - 010 = 002 - 010 = 006$
TEST 3	$ 2(374) - 004 = 2(-004) - 004 = 014$
TEST 4	$ 2(002) - 004 = 004 - 004 = 000$

PROGRAMMING AND DATA SHEET

1- (a) PROGRAM TEST SHEET

TEST	A	B	(A - B)	RESULT
1	200	100	100	✓
2	100	010	090	✓
3	010	000	010	✓
4	000	000	000	✓

CHECK THE DATA

TEST 1	$200 - 100 = 100$	$200 - 100 = 100$
TEST 2	$100 - 010 = 090$	$100 - 010 = 090$
TEST 3	$010 - 000 = 010$	$010 - 000 = 010$
TEST 4	$000 - 000 = 000$	$000 - 000 = 000$

ITERATIVE PROGRAM LOOPS

PROGRAM TEST $X = A \times B + C$

TEST RESULTS					
TEST	A	B	C	EXPECTED ANSWER	ACTUAL ANSWER
1	010	007	003	073	073
2	030	004	071	231	231
3	020	005	031	151	151
4	010	010	002	102	102
5	036	002	367(-11)	063	063

CHECK $X = A \cdot B + C$ (OCTAL)

TEST 2 $X = (030)(004) + 071$

```

  030
x 004
-----
 140
+ 071
-----
231 ANSWER

```

TEST 3 $X = (020)(005) + 031$

```

  020
x 005
-----
 120
+ 031
-----
151 ANSWER

```

TEST 4 $X = (010)(010) + 002$

```

  010
x 010
-----
 100
+ 002
-----
102 ANSWER

```

TEST 5 $X = (036)(002) + 367$

```

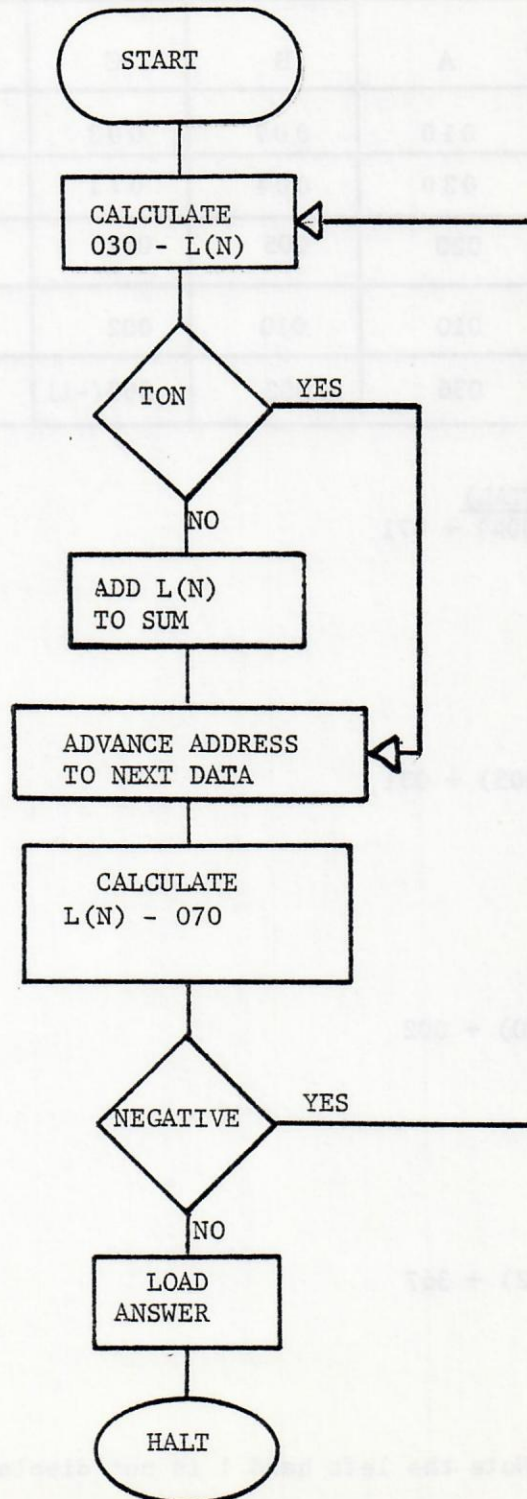
  036
x 002
-----
 074
+ 367
-----
063 ANSWER (Note the left hand 1 is not displayed)

```

ITERATIVE PROGRAM LOOPS

QUESTION 1

FLOW CHART



ITERATIVE PROGRAM LOOPS

QUESTION #1
SYMBOLIC PROGRAM

	WORD	COMMENT
	L(1) TRANS	TRANSFER TO L(1)
L(N1)	Ø	DATA
	Ø	
	Ø	
	Ø	
	Ø	
	Ø	
	Ø	
	Ø	ANSWER
L(1)	L(030) LOAD	LOAD 030
L(3)	L(N1) SUB	CALCULATE (030 - Ø)
	L(2) TON	TRANSFER ON NEGATIVE
	L(3) LOAD	LOAD L(3)
	L(N) ADD	CHANGE L(3) to L(4) LOAD INSTRUCTION
	L(4) STORE	STORE IN L(4)
L(4)	L(N1) LOAD	LOAD DATA
	L(ANS) ADD	ADD TO ANS
	L(ANS) STORE	STORE NEW ANSWER
L(2)	L(010) LOAD	LOAD 010
	L(3) ADD	ADD TO L(3) (ADVANCE DATA ADDRESS)
	L(3) STORE	STORE NEW INSTRUCTION IN L(3)
	L(100) SUB	CHECK FOR LAST DATA LOCATION
	L(1) TON	TRANSFER ON NEGATIVE TO L(1)
	L(ANS) LOAD	LOAD ANSWER
	L(NO) HALT	HALT

ITERATIVE PROGRAM LOOPS

SYMBOLIC PROGRAM

	WORD	COMMENT
L(100)	100	100
L(030)	030	030
L(101)	101	010
L(NO)	000	000
L(N)	002	002 (Modification of L(3)(SUB) to L(4) LOAD)

ITERATIVE PROGRAM LOOPS

SUM OF ALL NUMBERS UNDER 25

PROGRAM : Seven numbers are to be stored in Locations 01 thru 07 etc.

ADDRESS		WORD			COMMENTS
0	0	1	0	(1)	TRANSFER AROUND DATA TO 11
0	1	Ø	Ø	Ø	DATA
0	2	Ø	Ø	Ø	DATA
0	3	Ø	Ø	Ø	DATA
0	4	Ø	Ø	Ø	DATA
0	5	Ø	Ø	Ø	DATA
0	6	Ø	Ø	Ø	DATA
0	7	Ø	Ø	Ø	DATA
1	0	Ø	Ø	Ø	ANSWER
1	1	3	2	(6)	LOAD 030
1	2	0	1	(4)	CALCULATE 030 - Ø
1	3	2	1	(0)	TON TO 22
1	4	1	2	(6)	LOAD 12
1	5	3	5	(5)	ADD 002 (change instruction from SUB to LOAD)
1	6	1	7	(2)	STORE NEW INSTRUCTION
1	7	Ø	Ø	Ø	Ø ADDRESS LOAD INSTRUCTION
2	0	1	0	(5)	ADD TO ANSWER
2	1	1	0	(2)	STORE NEW ANSWER
2	2	3	3	(6)	LOAD 010
2	3	1	2	(5)	ADVANCE ADDRESS OF INSTRUCTION 12
2	4	1	2	(2)	STORE NEW INSTRUCTION
2	5	3	1	(4)	SUBTRACT 080
2	6	1	0	(0)	TON TO 11
2	7	1	0	(6)	LOAD ANSWER
3	0	3	4	(7)	HALT
3	1	1	0	0	100
3	2	0	3	0	030
3	3	0	1	0	010
3	4	0	0	0	000 (LNO)
3	5	0	0	2	020 (L(N)

ITERATIVE PROGRAM LOOPS

OPERATOR INSTRUCTIONS
LOAD DATA IN LOCATIONS 01 through 07
LOAD 000 IN LOCATION 10
LOAD 014 IN LOCATION 12
START PROGRAM AT 000
AUTOMATIC HALT
ANSWER IN DISPLAY AND LOCATION 10

PROGRAM TEST

TEST RESULTS			
TEST NO.	1	2	3
Ø 1	010	020	.370
Ø 2	001	010	010
Ø 3	001	031	010
Ø 4	035	062	040
Ø 5	035	035	020
Ø 6	001	001	030
Ø 7	001	001	020
RESULT	014	032	100

PROGRAMMING TECHNIQUES

PROCEDURE: (MINIMUM COMMANDS)

$$X = A + B - C$$

SYMBOLIC PROGRAM

	WORD	COMMENT
	L(1) TRA	TRANSFER TO L1
L(A)	Ø	A
L(B)	Ø	B
L(C)	Ø	C
L(1)	L(N) STORE	STORE ACCUMULATOR CONTENTS IN L(N)
	L(N) SUB	SUBTRACT L(N) FROM ACCUMULATOR ACCUMULATOR BECOMES ZERO
	L(A) SUB	ACCUMULATOR BECOMES -A
	L(-A) STORE	STORE -A in L(-A)
	L(-A) SUB	SUBTRACT (-A) FROM ACCUMULATOR ACCUMULATOR BECOMES ZERO
	L(B) SUB	SUBTRACT B (ACCUMULATOR BECOMES -B)
	L(-B) STORE	STORE -B IN L(-B)
	L(-B) SUB	SET ACCUMULATOR TO ZERO
	L(-B) SUB	ACCUMULATOR IS B
	L(-A) SUB	ACCUMULATOR IS $B - (-A) = B + A$
	L(C) SUB	ACCUMULATOR IS $A + B - C$
L(N)	Ø	HALT
L(-A)	Ø	-A
L(-B)	Ø	-B
L(NO)	000	ZERO

PROGRAMMING TECHNIQUES

PROGRAM : $X = A + B - C$					
ADDRESS		WORD			COMMENTS
0	0	0	3	(1)	TRANSFER TO 04
0	1	Ø	Ø	Ø	A
0	2	Ø	Ø	Ø	B
0	3	Ø	Ø	Ø	C
0	4	2	0	(2)	STORE CONTENTS OF ACCUMULATOR IN 20
0	5	2	0	(4)	SUBTRACT L(20) FROM ACCUM. (ACCUM. IS ZERO)
0	6	0	1	(4)	SUBTRACT A
0	7	2	1	(2)	STORE -A IN 21
1	0	2	1	(4)	SUB(-A) (SET ACCUMULATOR TO ZERO)
1	1	0	2	(4)	SUB (B)
1	2	2	2	(2)	STORE -B IN L(22)
1	3	2	2	(4)	SUB (-B) (ACCUMULATOR TO ZERO)
1	4	2	2	(4)	SUB (-B) IS B
1	5	2	1	(4)	SUB (-A), $B - (-A) = A + B$
1	6	0	3	(4)	SUB C
1	7	2	3	(7)	HALT (ACCUMULATOR HAS $A + B - C$)
2	0	Ø	Ø	Ø	
2	1	Ø	Ø	Ø	
2	2	Ø	Ø	Ø	
2	3	0	0	0	ZERO

PROGRAM TEST

$$X = A + B - C$$

TEST	A	B	C	X
1	004	007	005	006
2	010	012	003	017
3	021	013	032	002
4	034	005	042	377

PROGRAMMING TECHNIQUESTEST #1 (OCTAL) $X = 004 + 007 - 005$

004	A
+ 007	B
10	CARRY
013	
- 005	C
6	ANSWER (IN OCTAL)

TEST #2 (OCTAL) $X = 010 + 012 - 003$

010	A
+ 012	B
00	CARRY
022	
- 003	C
017	ANSWER (IN OCTAL)

TEST #3 (OCTAL) $X = 021 + 013 - 032$

021	A
+ 013	B
0	CARRY
034	
- 032	C
002	ANSWER (IN OCTAL)

TEST #4 (OCTAL) $X = 034 + 005 - 042$

034	A
+ 005	B
10	CARRY
041	
- 042	C
- 001	= 377 = ANSWER (IN OCTAL)

PROGRAMMING TECHNIQUES

PROCEDURE (BIT SELECTION)

SYMBOLIC PROGRAM (SELECTION OF MIDDLE DIGIT - 3 BITS)

	WORD	COMMENT
	L(1) TRANSFER	TRANSFER TO L(1)
L(0)	0	
L(1)	LOAD L(0)	LOAD 0
	ADD L(0)	CALCULATE 2 0
	STORE L(40)	STORE 2 0
	ADD L(40)	CALCULATE 4 0
	STORE L(40)	STORE 4 0
	ADD L(40)	CALCULATE 8 0
	STORE L(R)	STORE 8 0
	ADD L(R)	CALCULATE 16 0
	STORE L(R)	STORE 16 0
	ADD L(R)	CALCULATE 32 0
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	RESULTS IN RIGHT 3 BITS
	STORE L(R)	STORE RIGHT HAND 3 BITS
	LOAD L(40)	LOAD 4 0
	SHIFT RIGHT	
	SHIFT RIGHT	RIGHT HAND 6 BITS

SYMBOLIC PROGRAM (continued)

	WORD	COMMENT
	SUB L(R)	MIDDLE 3 BITS
	STORE (L(ANS))	
	L(NO) HALT	
L(4Ø)	Ø	4 Ø LOCATION
L(R)	Ø	RIGHT HAND 3 BITS
L(NO)	Ø	ZERO
L(ANS)	Ø	ANSWER

PROGRAMMING TECHNIQUES

PROGRAM : BIT SELECTION - MIDDLE DIGIT - 3 BITS

ADDRESS		WORD			COMMENTS
0	0	0	1	(1)	TRANSFER TO 02
0	1	0	0	0	DATA
0	2	0	1	(6)	LOAD 0
0	3	0	1	(5)	CALCULATE 20
0	4	3	0	(2)	STORE IN 30
0	5	3	0	(5)	CALCULATE 40
0	6	3	0	(2)	STORE IN 30
0	7	3	0	(5)	CALCULATE 80
1	0	3	1	(2)	STORE 80 IN 31
1	1	3	1	(5)	CALCULATE 160
1	2	3	1	(2)	STORE 160 IN 31
1	3	3	1	(5)	CALCULATE 320
1	4	0	0	(3)	SHIFT RIGHT
1	5	0	0	(3)	SHIFT RIGHT
1	6	0	0	(3)	SHIFT RIGHT
1	7	0	0	(3)	SHIFT RIGHT
2	0	0	0	(3)	SHIFT RIGHT RESULTS IN RT. HAND 3 BITS ONLY
2	1	3	1	(2)	STORE IN 31
2	2	3	0	(6)	LOAD 40
2	3	0	0	(3)	SHIFT RIGHT
2	4	0	0	(3)	SHIFT RIGHT, 0 with 2 LEFT BITS AT ZERO
2	5	3	1	(4)	REMOVE RIGHT 3 BITS
2	6	3	3	(2)	STORE IN 33
2	7	3	2	(7)	HALT
3	0	0	0	0	L(40)
3	1	0	0	0	L(R)
3	2	0	0	0	L(NO)
3	3	0	0	0	L(ANS)

PROGRAMMING TECHNIQUES

OPERATOR INSTRUCTIONS

ENTER DATA IN ADDRESS 01

START AT LOCATION 00

AUTOMATIC STOP

ANSWER APPEARS IN DISPLAY AND IN LOCATION 33

BIT SELECTION 		
TEST	INPUT	OUTPUT
1	136	030
2	264	060
3	252	050
4	123	020
5	147	040

PROGRAMMING TECHNIQUESQUESTION #2 BCD TO BINARY CONVERSIONSYMBOLIC PROGRAM

	WORD	COMMENT
L(1)	L(2) TRA	TRANSFER TO L(2)
L(0)	0	
L(2)	LOAD L(0)	LOAD 0
	ADD L 0	CALCULATE 20
	STORE L(X)	STORE 20 IN L(X)
	ADD L(X)	CALCULATE 40
	STORE L(X)	STORE 40 IN L(X)
	ADD L(X)	CALCULATE 80
	STORE L(X)	STORE 80 IN L(X)
	ADD L(X)	CALCULATE 160
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	RIGHT HAND 4 BITS ONLY
	STORE L(X)	RIGHT DIGIT STORED IN L(X)
	LOAD L(0)	LOAD 0
	SUB L(X)	REMOVE RIGHT HAND 4 BITS
	SHIFT RIGHT	
	STORE IN L(N)	
	SHIFT RIGHT	
	SHIFT RIGHT	
	ADD L(N)	LEFT DIGIT x 10
	ADD L(X)	BINARY EQUIVALENT
	L(ANS) STORE	STORE ANSWER IN L(ANS)
	HALT	
L(NO)	0	ZERO
L(X)	0	
L(N)	0	
L(ANS)	0	ANSWER

PROGRAMMING TECHNIQUES

PROGRAM : BCD TO BINARY CONVERSION					
ADDRESS		WORD			COMMENTS
0	0	0	1	(1)	TRANSFER TO 02
0	1	0	0	0	DATA
0	2	0	1	(6)	LOAD 0
0	3	0	1	(5)	ADD 0
0	4	3	2	(2)	STORE 2 0 IN 31
0	5	3	2	(5)	CALCULATE 4 0
0	6	3	2	(2)	STORE 4 0 IN 31
0	7	3	2	(5)	CALCULATE 8 0
1	0	3	2	(2)	STORE 8 0 IN 31
1	1	3	2	(5)	CALCULATE 16 0
1	2	0	0	(3)	SHIFT RIGHT
1	3	0	0	(3)	SHIFT RIGHT
1	4	0	0	(3)	SHIFT RIGHT
1	5	0	0	(3)	SHIFT RIGHT (ACCUM. SHOWS ONLY RT. HAND 4BITS)
1	6	3	2	(2)	STORE RIGHT DIGIT IN 31
1	7	0	1	(6)	LOAD 0
2	0	3	2	(4)	SUBTRACT 31 - RESULTS IN LEFT HAND DIGIT
2	1	0	0	(3)	SHIFT RIGHT (MULTIPLY BY 8)
2	2	3	3	(2)	STORE IN 32
2	3	0	0	(3)	SHIFT RIGHT
2	4	0	0	(3)	SHIFT RIGHT (MULTIPLY BY 2)
2	5	3	3	(5)	ADD N x 8
2	6	3	2	(5)	ADD RIGHT HAND BITS
2	7	3	4	(2)	STORE ANSWER IN 34
3	0	3	1	(7)	HALT
3	1	0	0	0	000
3	2	0	0	0	L(X)
3	3	0	0	0	L(N)
3	4	0	0	0	L(ANS)

PROGRAMMING TECHNIQUES

OPERATOR INSTRUCTIONS

LOAD 0 INTO 01 (BCD NUMBER WRITTEN IN OCTAL)

START PROGRAM AT 00

AUTOMATIC STOP

ANSWER IN LOCATION 34 and on

DISPLAY IN OCTAL

BCD TO BINARY		
TEST	BCD (CODE)	BINARY (OCTAL)
1	$\begin{array}{ccccccc} & 4 & & & 9 & & \\ \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} \\ & 1 & & & 1 & & & 1 \end{array}$	$\begin{array}{ccc} 0 & 6 & 1 \\ 00 & 110 & 001 \\ 32 + 16 + 1 = & 49 \end{array}$
2	$\begin{array}{ccccccc} & 2 & & & 1 & & \\ \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{1} \\ & 0 & & & 4 & & & 1 \end{array}$	$\begin{array}{ccc} 0 & 2 & 5 \\ 00 & 010 & 101 \\ 16 + 4 + 1 = & 21 \end{array}$
3	$\begin{array}{ccccccc} & 9 & & & 6 & & \\ \underline{1} & \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{1} & \underline{1} & \underline{0} \\ & 2 & & & 2 & & & 6 \end{array}$	$\begin{array}{ccc} 1 & 4 & 0 \\ 01 & 100 & 000 \\ 64 + 32 = & 96 \end{array}$
4	$\begin{array}{ccccccc} & 2 & & & 7 & & \\ \underline{0} & \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{1} & \underline{1} & \underline{1} \\ & 0 & & & 4 & & & 7 \end{array}$	$\begin{array}{ccc} 0 & 3 & 3 \\ 00 & 011 & 011 \\ 16 + 8 + 2 + 1 = & 27 \end{array}$
5	$\begin{array}{ccccccc} & 3 & & & 4 & & \\ \underline{0} & \underline{0} & \underline{1} & \underline{1} & \underline{0} & \underline{1} & \underline{0} & \underline{0} \\ & 0 & & & 6 & & & 4 \end{array}$	$\begin{array}{ccc} 0 & 4 & 2 \\ 00 & 100 & 010 \\ 32 + 2 = & 34 \end{array}$

PROGRAMMING TECHNIQUESQUESTION #3 BINARY TO BCD CONVERSIONSYMBOLIC PROGRAM

	WORD	COMMENT
	L(2) TRA	TRANSFER TO L(2)
L(0)	0	BINARY NUMBER
L(2)	LOAD L(NO)	
	STORE IN L(4)	SET LOCATION L(4) TO ZERO
L(5)	LOAD L(0)	LOAD BINARY NUMBER
	SUBTRACT L(10)	SUBTRACT 10
	TRANS ON NEG TO L(3)	TRANSFER ON NEGATIVE TO L(3)
	STORE IN L(0)	0 -10 is STORED
	LOAD L(001)	LOAD 1
	ADD L(4)	ADD 1 TO L(4)
	STORE L(4)	STORE L(4) + 1 IN L(4)
	TRANSFER TO L(5)	REPEAT SUBTRACTION
L(3)	STORE IN L(N)	STORE REMAINDER MINUS 10
	LOAD L(4)	LOAD NUMBER OF 10's
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	STORE IN L(4)	NUMBER OF 10's IN LEFT HAND DIGIT
	LOAD L(N)	REMAINDER MINUS 10
	ADD L(10)	CALCULATE REMAINDER
	ADD L(4)	ADD 10's DIGIT TO 1's DIGIT
	STORE IN L(ANS)	STORE ANSWER
	L(NO) HALT	HALT
L(NO)	0	ZERO
L(N)	0	LOCATION OF REMAINDER
L(ANS)	0	ANSWER
L(10)	10	LOCATION OF 10
L(4)	0	LOCATION OF NUMBER OF 10's
L(001)	001	

PROGRAMMING TECHNIQUES

PROGRAM : BINARY TO BCD CONVERSION					
ADDRESS		WORD			COMMENTS
0	0	0	1	(1)	TRANSFER TO 02
0	1	0	0	0	BINARY NUMBER
0	2	3	0	(6)	LOAD ZERO (INITIALIZE L(4) TO ZERO)
0	3	3	4	(2)	STORE ZERO IN 34
0	4	0	1	(6)	LOAD 0
0	5	3	3	(4)	SUBTRACT 10
0	6	1	3	(0)	TON TO 14 ON NEGATIVE
0	7	0	1	(2)	STORE 0 - 10 IN 01
1	0	3	5	(6)	LOAD 001
1	1	3	4	(5)	ADD L(4) (COUNTING 10's)
1	2	3	4	(2)	STORE RESULT IN 34 (L(4))
1	3	0	3	(1)	TRANSFER TO 04
1	4	3	1	(2)	STORE REMAINDER MINUS 10
1	5	3	4	(6)	LOAD NUMBER OF 10's
1	6	0	0	(3)	SHIFT RIGHT
1	7	0	0	(3)	SHIFT RIGHT
2	0	0	0	(3)	SHIFT RIGHT
2	1	0	0	(3)	SHIFT RIGHT
2	2	3	4	(2)	STORE LEFT HAND DIGIT INTO 34
2	3	3	1	(6)	LOAD REMAINDER MINUS 10
2	4	3	3	(5)	ADD 10 (CALCULATE REMAINDER)
2	5	3	4	(5)	ADD 10's DIGIT
2	6	3	2	(2)	STORE ANSWER
2	7	3	0	(7)	HALT
3	0	0	0	0	L(NO)
3	1	0	0	0	LOCATION OF REMAINDER
3	2	0	0	0	L(ANS)
3	3	0	1	2	L(10) (BINARY)
3	4	0	0	0	L(4)
3	5	0	0	1	L(001)

PROGRAMMING TECHNIQUES

OPERATORS INSTRUCTIONS

STORE 0 IN LOCATION 01

START AT 00

AUTOMATIC STOP

ANSWER WILL APPEAR IN ACCUMULATOR AND WILL

BE STORED IN 32

PROGRAM TEST

TEST	BINARY NUMBER	OCTAL DISPLAY	BCD NO.	OCTAL DISPLAY WILL READ	CHECK
1	00100101	045	3 7 0011 0111	0 6 7 00 110 111	✓
2	00011001	031	2 5 0010 0101	0 4 5 00 100 101	✓

QUESTION #4

Write a program to find the quotient of two numbers, and the remainder. Set the program up so that all storage spaces are properly initialized and the program can be run several times without entering new data. When the program halts the quotient will be display. When the start button is pushed again the remainder will be displayed.

PROGRAMMING TECHNIQUESSYMBOLIC PROGRAM

	WORD	COMMENT
	TRANS TO L(1)	JUMP OVER DATA
L(N)	Ø	NUMERATOR
L(D)	Ø	DENOMINATOR
L(1)	LOAD L(NO)	LOAD ZERO
	STORE L(R)	SET L(R) REMAINDER TO ZERO
	STORE L(Q)	SET L(Q) QUOTIENT TO ZERO
	LOAD L(N)	LOAD NUMERATOR
	STORE L(S)	STORE IN L(S)
L(2)	L(S) LOAD	LOAD L(S)
	SUB L(D)	SUBTRACT DENOMINATOR
	STORE IN L(S)	STORE RESULT IN L(S)
	TRA ON NEG L(3)	TRANSFER ON NEGATIVE
	LOAD L(001)	LOAD 1
	ADD L(Q)	ADD TO L(Q) QUOTIENT
	STORE IN L(Q)	STORE QUOTIENT
	TRA TO L(2)	TRANSFER TO L(2)
L(3)	LOAD L(S)	
	ADD L(D)	CALCULATE REMAINDER
	STORE L(R)	STORE REMAINDER
	LOAD L(Q)	LOAD ANSWER
	HALT	
	LOAD L(R)	LOAD REMAINDER
	HALT	
L(NO)	000	
L(S)	Ø	
L(R)	Ø	
L(Q)	Ø	
L(001)	001	

PROGRAMMING TECHNIQUES

PROGRAM : DIVIDE - DISPLAY QUOTIENT & REMAINDER

ADDRESS		WORD			COMMENTS
0	0	0	2	(1)	TRANSFER TO 03
0	1	Ø	Ø	Ø	NUMERATOR
0	2	Ø	Ø	Ø	DENOMINATOR
0	3	2	7	(6)	LOAD ZERO
0	4	3	1	(2)	SET 31 TO ZERO
0	5	3	2	(2)	SET 32 TO ZERO
0	6	0	1	(6)	LOAD NUMERATOR
0	7	3	0	(2)	STORE IN 30
1	0	3	0	(6)	LOAD 30
1	1	0	2	(4)	SUBTRACT DENOMINATOR
1	2	3	0	(2)	STORE RESULT IN 27 (L(S))
1	3	1	7	(0)	TON TO 20 (L(3))
1	4	3	3	(6)	LOAD 001
1	5	3	2	(5)	ADD TO QUOTIENT
1	6	3	2	(2)	STORE IN L(Q) QUOTIENT
1	7	0	7	(1)	TRANSFER TO 10
2	0	3	0	(6)	LOAD L(S) (L(30))
2	1	0	2	(5)	ADD DENOMINATOR (REMAINDER)
2	2	3	1	(2)	STORE REMAINDER IN 31
2	3	3	2	(6)	LOAD QUOTIENT
2	4	2	7	(7)	HALT
2	5	3	1	(6)	LOAD REMAINDER
2	6	2	7	(7)	HALT
2	7	0	0	0	
3	0	Ø	Ø	Ø	
3	1	Ø	Ø	Ø	
3	2	Ø	Ø	Ø	
3	3	0	0	1	

PROGRAMMING TECHNIQUES

OPERATOR INSTRUCTIONS

LOAD NUMERATOR INTO LOCATION 01

LOAD DENOMINATOR INTO LOCATION 02

START AT 00

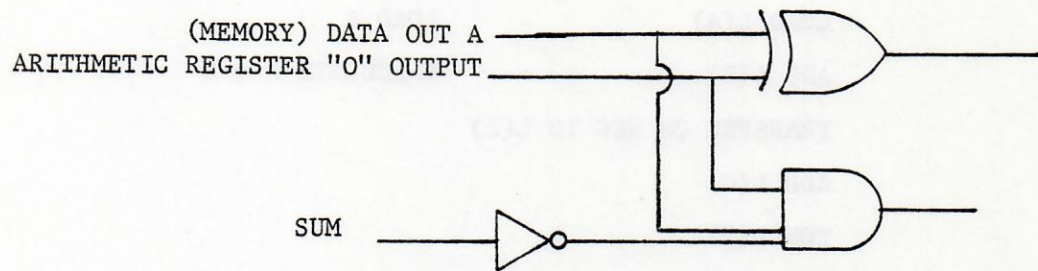
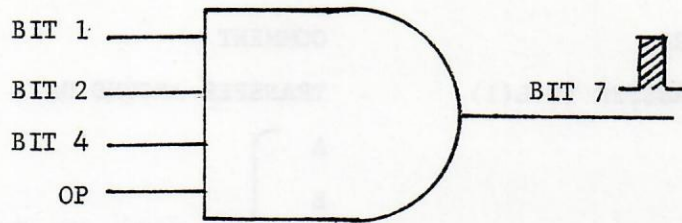
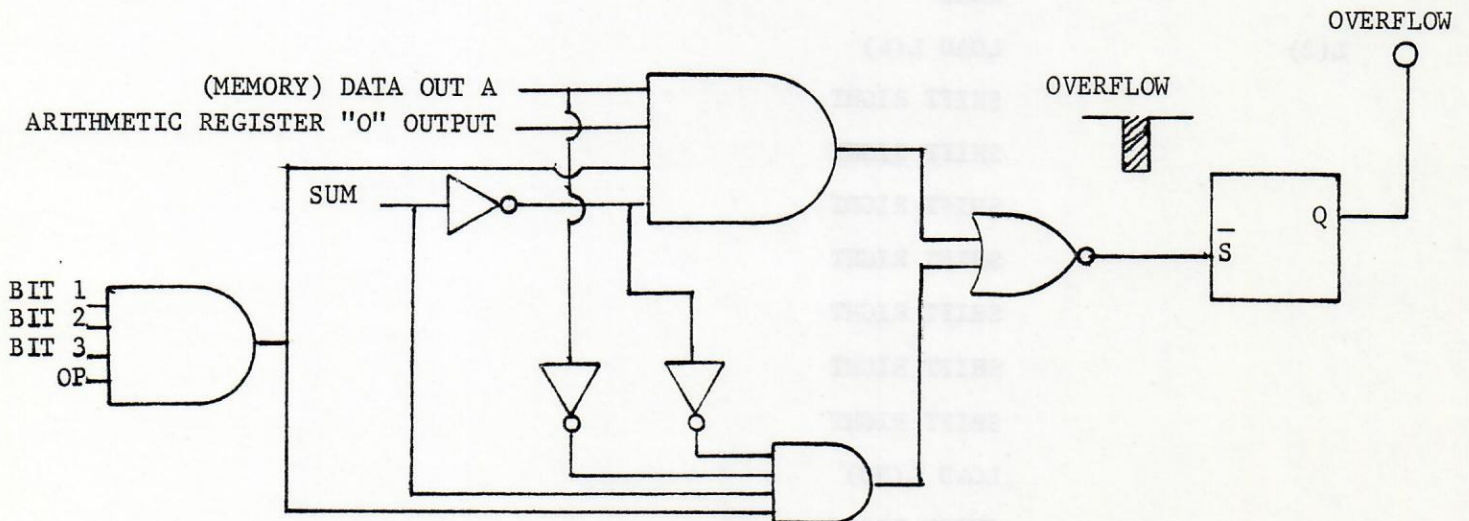
QUOTIENT WILL APPEAR ON DISPLAY AND IN LOCATION 31

PUSH START BUTTON

REMAINDER WILL APPEAR ON DISPLAY AND IN LOCATION 30

PROGRAM TEST

TEST	NUMERATOR	DENOMINATOR	QUOTIENT	REMAINDER
1	4	2	2	0
2	5	2	2	1
3	2	3	0	2
4	21	4	4	1
5	174	3	51	1

PROGRAMMING TECHNIQUESQUESTION #5

PROGRAMMING TECHNIQUESQUESTION #6

$X = A + B + C + D$ with positive numbers

OVERFLOW DETECTION
SYMBOLIC PROGRAM

	WORD	COMMENT
	TRANSFER TO L(1)	TRANSFER AROUND DATA
L(A)	Ø	A
L(B)	Ø	B
L(C)	Ø	C
L(D)	Ø	D
		} NOTE: ALL POSITIVE NUMBERS
L(1)	LOAD L(A)	LOAD A
	ADD L(B)	CALCULATE A + B
	TRANSFER ON NEG TO L(2)	
	ADD L(C)	
	TON L(2)	
	ADD L(D)	
	TON L(2)	
	STORE IN L(3)	
	HALT	
L(2)	LOAD L(4)	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	LOAD L(NO)	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	SHIFT RIGHT	
	TRA TO L(2)	

COMMENT

PROGRAMMING TECHNIQUESPROGRAM : $X = A + B + C + D$ with OVERFLOW ALARM

ADDRESS		WORD			COMMENTS
0	0	0	4	(1)	TRANSFER AROUND DATA TO 05
0	1	0	0	0	A
0	2	0	0	0	B
0	3	0	0	0	C
0	4	0	0	0	D
0	5	0	1	(6)	LOAD A
0	6	0	2	(5)	ADD B, CALCULATE A + B
0	7	1	5	(0)	TON TO 16 CHECK FOR OVERFLOW
1	0	0	3	(5)	CALCULATE A + B + C
1	1	1	5	(0)	TON TO 16
1	2	0	4	(5)	CALCULATE A + B + C + D
1	3	1	5	(0)	TON TO 16 CHECK FOR OVERFLOW
1	4	3	5	(2)	STORE X in 35
1	5	3	7	(7)	HALT
1	6	3	6	(6)	LOAD 377
1	7	0	0	(3)	SHIFT RIGHT
2	0	0	0	(3)	SHIFT RIGHT
2	1	0	0	(3)	SHIFT RIGHT
2	2	0	0	(3)	SHIFT RIGHT
2	3	0	0	(3)	SHIFT RIGHT
2	4	0	0	(3)	SHIFT RIGHT
2	5	0	0	(3)	SHIFT RIGHT
2	6	3	7	(6)	LOAD L(NO)
2	7	0	0	(3)	SHIFT RIGHT
3	0	0	0	(3)	SHIFT RIGHT
3	1	0	0	(3)	SHIFT RIGHT
3	2	0	0	(3)	SHIFT RIGHT
3	3	0	0	(3)	SHIFT RIGHT
3	4	1	5	(1)	TRANSFER TO 16
3	5	0	0	0	
3	6	3	7	7	
3	7	0	0	0	

PROGRAMMING TECHNIQUES

OPERATOR INSTRUCTIONS

ENTER A IN 01 B IN 02 C IN 03 AND D in 04

START AT 00

AUTOMATIC HALT

ANSWER IN DISPLAY AND LOCATION 35

IF OVERFLOW DISPLAY WILL ALTERNATE BETWEEN 377 & 0

PROGRAM TEST

TEST	A	B	C	D	X	OVERFLOW
1	002	004	006	010	024	NO
2	102	042	016	020	-	YES
3	100	100	100	100	-	YES
4	076	034	020	001	153	NO

PROGRAMMING TECHNIQUESPROCEDURE: SUB ROUTINES

NOTE: This program uses 41 (Octal) Memory Locations. The Symbolic Program is missing the steps 03 and 12, which were omitted by the person who wrote the Symbolic Program. This happens in practical situations and must be correct by the programmer.

In this case, we have an additional problem, with the 2 additional steps, the program exceeds the memory space available. In a case like this the programmer must reorganize the program and economize on the amount of memory used.

One solution to placing this program in 37 (OCTAL) Memory Locations is given below.

SYMBOLIC PROGRAM $X = |A| + B - 2 |C|$

	WORD	COMMENT
	L 11 LOAD	LOAD EXIT INSTRUCTION
	L 6 STORE	STORE EXIT INSTRUCTION
	L 17 LOAD	LOAD A
	L(1) TRA	TRANSFER TO MAG SUB ROUTINE
L 12	L(3) LOAD	LOAD ANSWER
	L(17) STORE	STORE IN A
	L(16) LOAD	LOAD EXIT INSTRUCTION
	L(6) STORE	STORE EXIT INSTRUCTION
	L(19) LOAD	LOAD C
	L(1) TRA	TRANSFER TO MAG
L 14	L(17) LOAD	LOAD $ A $
	L(18) ADD	CALCULATE $ A + B$
	L(3) SUB	CALCULATE $ A + B - C $
	L(3) SUB	CALCULATE $ A + B - 2 C $
	L(15) STORE	STORE ANSWER
	L(NO) HALT	HALT
L(11)	L(12) TRA	EXIT
L(16)	L(14) TRA	EXIT
L(17)	Ø	A
L(18)	Ø	B
L(19)	Ø	C
L(15)	X	ANSWER
↙ MAG SUB ROUTINE		
L(1)	L(3) STORE	STORE A IN L(3)

PROGRAMMING TECHNIQUES

	WORD	COMMENT
	L(2) TON	TRANSFER ON NEGATIVE TO L(2)
L(6)	TRANSFER	EXIT TRANSFER
L(2)	L(NO) LOAD	LOAD ZERO
	L(3) SUB	SUBTRACT A
↑ END MAG SUB ROUTINE	L(1) TRA	TRANSFER TO L(1)
L(NO)	L(NO)	ZERO
L(3)	Ø	
	SPARE	
	SPARE	

PROGRAMMING TECHNIQUES

The Symbolic Program given in the experiment results in the following CODED PROGRAM.

PROGRAM : $ A + B - 2 C $					
ADDRESS		WORD			COMMENTS
0	0	2	3	(6)	LOAD A
0	1	3	0	(2)	STORE IN 30 (IN MAG ROUTINE)
0	2	2	1	(6)	LOAD 21 (EXIT INSTRUCTION)
0	3	3	2	(2)	STORE IN 32 (EXIT FROM MAG)
0	4	3	0	(1)	TRANSFER TO MAG.
0	5	4	0	(6)	LOAD $ A $
0	6	2	7	(2)	STORES ANSWER L 13 $\rightarrow A $
0	7	2	5	(6)	SUB ROUTINE ENTRY FOR C
1	0	3	6	(2)	STORE C IN MAG.
1	1	2	2	(6)	LOAD EXIT INSTRUCTION
1	2	3	2	(2)	STORE EXIT INSTRUCTION IN MAG.
1	3	2	7	(1)	TRANSFER TO MAG.
1	4	2	7	(6)	LOAD $ A $
1	5	2	4	(5)	CALCULATE $ A + B$
1	6	4	0	(4)	CALCULATE $ A + B - C $
1	7	4	0	(4)	CALCULATE $ A + B - 2 C $
2	0	2	6	(2)	STORE ANSWER
2	1	0	4	(1)	TRANSFER TO 05
2	2	1	3	(1)	TRANSFER TO 14
2	3	Ø	Ø	Ø	A
2	4	Ø	Ø	Ø	B
2	5	Ø	Ø	Ø	C
2	6	Ø	Ø	Ø	X
2	7	Ø	Ø	Ø	$ A $
3	0	3	6	(6)	LOAD A
3	1	3	3	(0)	TON TO 34
3	2	4	1	(2)	STORE ANSWER
3	3	Ø	Ø	Ø	EXIT
3	4	4	0	(6)	LOAD ZERO
3	5	3	7	(4)	SUB A
3	6	3	1	(1)	TRANSFER TO 32
3	7	Ø	Ø	Ø	A
4	0	0	0	0	L(NO)
4	1	Ø	Ø	Ø	$ A $

PROGRAMMING TECHNIQUESPROGRAM TEST

$X = A + B - 2 C $				
TEST	A	B	C	X
1	1	4	1	3
2	-1 = 377	4	-1 = 377	3
3	-14 = 364	-2 = 376	-4 = 374	02

PROGRAMMING TECHNIQUESQUESTION #7

$$Y = \frac{A \cdot B + 17}{C + 4}$$

SUB ROUTINES FOR MULTIPLICATION & DIVISION

MULTIPLICATION SUB ROUTINE

L(1) START

L(P) DATA

L(M) DATA

L(2) EXIT INSTRUCTION

L(N) PRODUCT

DIVISION SUB ROUTINE

L(3) START

L(T) NUMERATOR

L(D) DENOMINATOR

L(4) EXIT INSTRUCTION

L(DIV) PRODUCT

PROGRAMMING TECHNIQUESMAIN ROUTINE (SYMBOLIC PROGRAM)

	WORD	COMMENT
	L(A) LOAD	LOAD A
	L(P) STORE	STORE IN MULT
	L(B) LOAD	LOAD B
	L(M) STORE	STORE IN MULT
	L(5) LOAD	TRANSFER TO L(6) EXIT INSTRUCTION
	L(2) STORE	STORE EXIT INSTRUCTION
	L(1) TRANSFER	TRANSFER TO MULT
L(6)	LOAD L(N)	
	STORE IN L(9)	STORE A·B IN L(9)
	ADD L(17)	CALCULATE A · B + 17
	STORE IN L(17)	STORE A · B + 17
	LOAD L(C)	LOAD C
	ADD L(N4)	CALCULATE C + 4
	L(D) STORE	STORE IN DENOMINATOR OF DIV SUB ROUTINE
	L(17) LOAD	LOAD A · B + 17
	L(T) STORE	STORE IN NUMERATOR OF DIV
	L(7) LOAD	EXIT INSTRUCTION (TRA TO L(8))
	L(4) STORE	STORE EXIT INSTRUCTION IN DIV
L(8)	LOAD L(DIV)	
	L(ANS) STORE	STORE ANSWER
	HALT	
L(5)	L(6) TRANSFER	
L(7)	L(8) TRANSFER	
L(A)	A	
L(B)	B	
L(17)	17	
L(C)	C	
L(N4)	004	
L(9)	Ø	
L(ANS)	ANSWER	

PROGRAMMING TECHNIQUESQUESTION #8

$$a) \quad X = A + B^2$$

$$X = A + (B ** 2)$$

$$b) \quad Y = (A + B)^2$$

$$Y = (A + B) ** 2$$

$$c) \quad Y = 3 \cos \left(\frac{A + B}{3} \right)$$

$$Y = 3 \cdot \text{COSINE} \left((A + B) / 3 \right)$$

$$d) \quad Z = \frac{\sqrt{-A \pm B^2 - 4AC}}{2A}$$

$$Z = (-A + (B ** 2) - (4 * A * C) ** (\frac{1}{2})) / (2 * A)$$

$$Z = (-A - (B ** 2) - (4 * A * C) ** (\frac{1}{2})) / (2 * A)$$

PRINTED CIRCUITS AND INTEGRATED CIRCUITS

EXP. 13-1 I.C. NAND Element

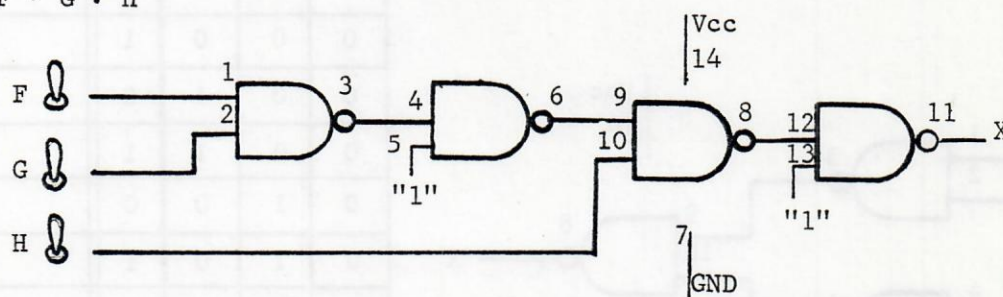
INPUT			OUTPUT		CHECK
F	G	H	6	7	
0	0	0	1	1	✓
0	0	1	1	0	✓
0	1	0	1	1	✓
0	1	1	1	0	✓
1	0	0	1	1	✓
1	0	1	1	0	✓
1	1	0	0	1	✓
1	1	1	0	1	✓

INSTRUCTORS NOTE:

Depending upon the scope of the program and the time available, the student may be required by the Instructor to generate a parts layout diagram, bill of materials, wiring diagram, etc on any job using the Ed-Lab 101 Prototype Board.

EXERCISE:

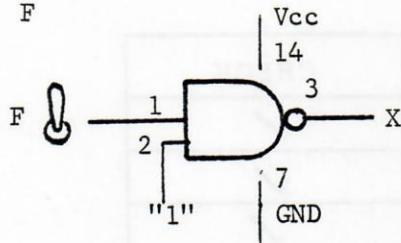
1(a) $X = F \cdot G \cdot H$



F	G	H	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

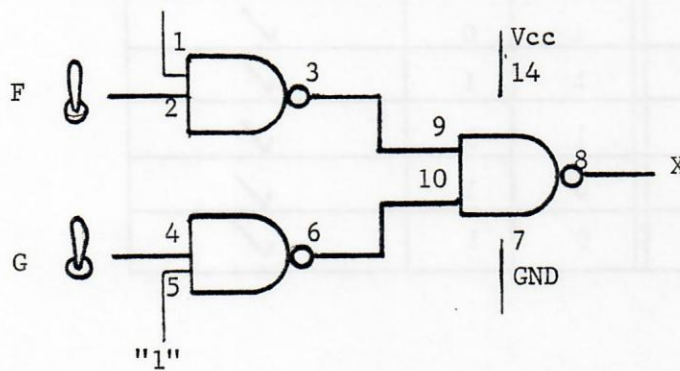
PRINTED CIRCUITS AND INTEGRATED CIRCUITS

1(b) $X = \overline{F}$



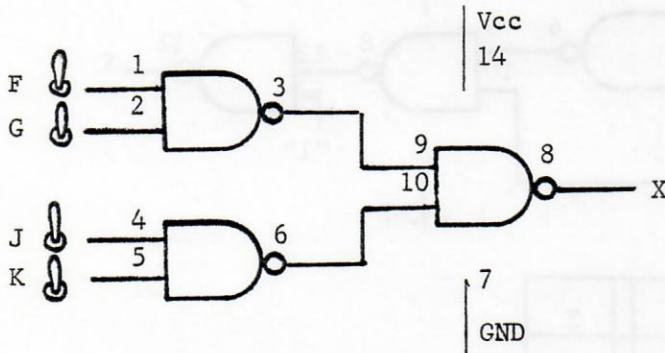
F	Y
0	1
1	0

1(c) $X = F + G$



F	G	X
0	0	0
0	1	1
1	0	1
1	1	1

1(d) $X = FG + JK$



F	G	J	K	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

PRINTED CIRCUITS AND INTEGRATED CIRCUITS

EXERCISE 2 & 3: These are exercises intended to be developed by the student and instructor with the components and the P.C. Cards available.

DIGITAL OUTPUT REFRESHER

Exp. 14-1 Digital Output Refresher

TEST	DATA TRANSMITTED	OUTPUT
1	100100111 ^{FIRST}	93
2	011001001	64
3	000000001	00
4	111111111	(BLANK)
5	001110001	38
6	011000101	62
7	010001111	47

NOTE: {

NOTE: This data is devised by the student and will be different for each students report.

DIGITAL REPRESENTATION

Note: Adjust the 1 bit so that the computer output reads as "1" when a "0" is transmitted and vice versa "0" when a "1" is transmitted.

This flip-flop is set to a "1" when a "0" is transmitted. This flip-flop is "cleared" to a "0" after the desired shift value from flip-flop # is completed.

The hexadecimal function properly.

TEST DATA

TEST	TRANSMITTED WORD	RECEIVED WORD
1	1010	1010
2	1101	1101
3	1011	1011

Enc. 12-4 Encoded Hexadecimal Data

INSTRUCTIONS

The word transmitted must be a 4 bit number with a 1 as the first bit. The following 3 bits are arranged in BCD with the least significant bit first. No change in the output number unless on the 4th display will occur until the last bit is received.

DIGITAL COMMUNICATIONS

Exp. 15-1 Transmit Data Register

Depressing and releasing the M Switch 4 times clears the register to 0000. The output lamp 5 indicates the shift of the data one place to the right.

PULSE AMPLITUDE MODULATION

When the data bit to be transmitted is a zero, the meter will indicate a pulse amplitude of 2.5 volts during the time of the pulse transmission.

When the data bit to be transmitted is a One the meter indicates a pulse amplitude of 3.5 volts during the pulse transmission.

FSK Modulator

When the data is a "0" the frequency output is lower than when the data is "1". The output frequency is high during transmission of a "1" and low during transmission of a "0".

Exp. 15-2 PDM Modulator-Transmitter

TEST	DATA SET					TRANSMITTER				
						L = LONG S = SHORT				
	A	B	C	D	E	5	4	3	2	1
1	1	0	1	0	1	L	S	L	S	L
2	1	0	0	1	1	L	S	S	L	L
3	1	1	0	1	0	L	L	S	L	S
4	1	0	1	1	0	L	S	L	L	S
5	1	1	0	1	0	L	L	S	L	S

Note: The difference in the pulse width for a "1" and a "0" is noticeable.

DIGITAL COMMUNICATIONS

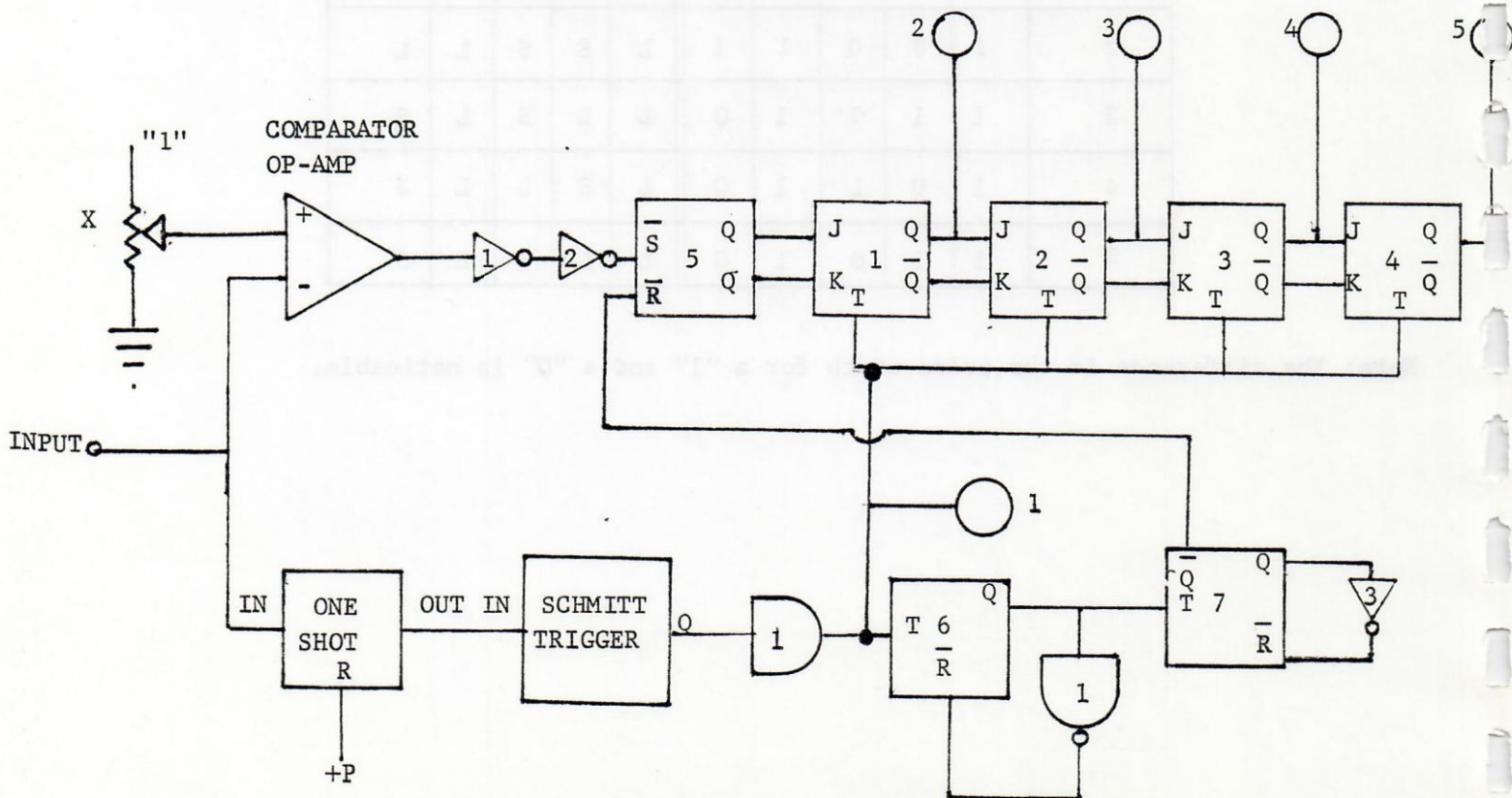
Exp. 15-3 PDM DECODER-PDM MODEM

TEST	TRANSMIT					RECEIVE				
	A	B	C	D	E	1	2	3	4	5
1	1	0	1	1	0	1	0	1	1	0
2	0	0	1	0	0	0	0	1	0	0
3	1	1	0	1	1	1	1	0	1	1
4	0	1	0	1	0	0	1	0	1	0
5	1	0	1	0	1	1	0	1	0	1

Note: The M Button must not be pushed in very rapid succession. The pulses must have time to complete before the new data bit is transmitted.

Questions:

- Using the PAM transmitter on page 5.



PAM (PULSE AMPLITUDE MODULATION) DEMODULATOR

DIGITAL COMMUNICATIONS

Note: Adjust the X POT so that the comparator output remains at "1" when a "0" is transmitted and goes to a "0" when a "1" is transmitted.

Flip-Flop 5 is set to a "1" when a "1" is transmitted. Flip-Flop 5 is "Cleared" to a "0" after the delayed shift pulse from Flip-Flop 6 is completed.

2. The Demodulator functions properly.

MODEM TEST

TEST	TRANSMITTED WORD	RECEIVED WORD
1	1 0 1 0	1 0 1 0
2	1 1 0 0	1 1 0 0
3	1 0 1 1	1 0 1 1

Exp. 15-4 Remote Display SystemINSTRUCTORS NOTE:

The word transmitted must be a 9 bit number with a 1 as the first bit. The following 8 bits are arranged in BCD with the least significant bit first. No change in the output number noted on the 900 display will occur until the last bit is received.

DIGITAL OUTPUT RESPONSE

Exp. 1A-1 Digital Output Response

TEST	DATA TRANSMITTED	OUTPUT
1	100100111	80
2	011001001	64
3	000000001	60
4	111111111	(BLANK)
5	001100001	56
6	011000101	62
7	010001111	63

NOTE:

NOTE: This data is derived by the student and will be different for each student's report.

NUMERICAL CONTROL SYSTEM

EXP. 16-1 SEQUENTIAL MEMORY

DATA TABLE

ADDRESS	DATA	
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	

ADDRESS	DATA	
20	20	
21	21	
22	22	
23	23	
24	24	
25	25	
26	26	
27	27	
30	30	
31	31	
32	32	
33	33	
34	34	
35	35	
36	36	
37	37	

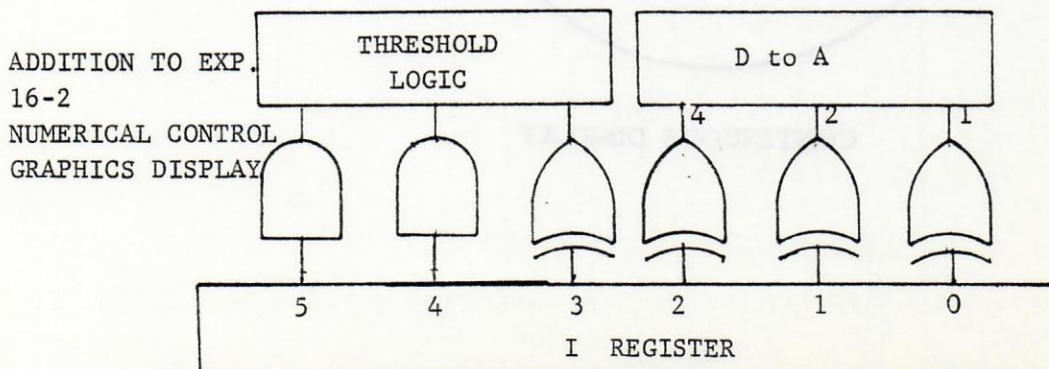
NOTE: The data stored in each location is made up by the student and will be different for each students report.

PROCEDURE (GRAPHICS DISPLAY)

EXP. 16-2 NUMERICAL CONTROL GRAPHICS DISPLAY

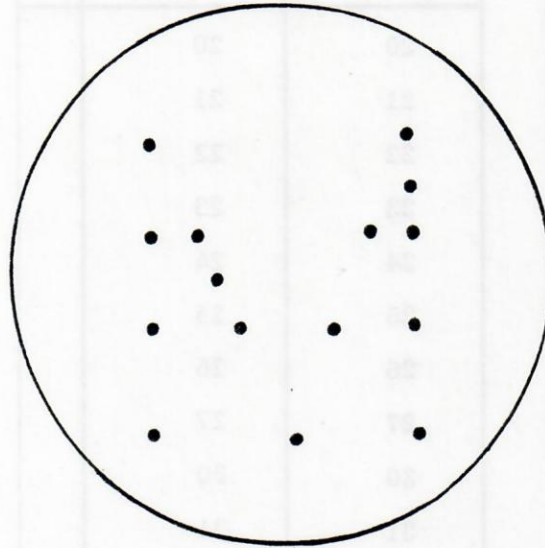
INSTRUCTOR'S NOTE:

Any ripple on the Ed-Lab 900 supply may result in a blurring or streaking of the programmed points on the oscilloscope screen. To eliminate this effect, use a buffer between each output from the "I" REGISTER and the inputs to the D to A Converter and the Threshold Logic element. Use AND's #3 and #4 and the 4 Exclusive OR gates as buffer elements.

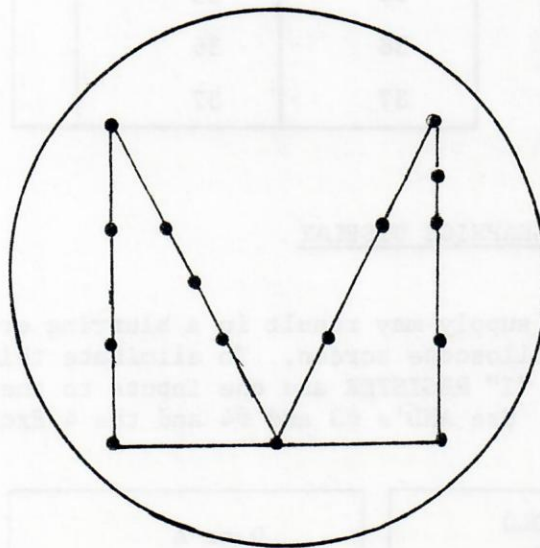


These Exclusive OR elements on the Ed-Lab 700 are used as buffers to eliminate any blurring due to ripple

NUMERICAL CONTROL SYSTEM



POINT BY POINT DISPLAY



CONTINUOUS DISPLAY

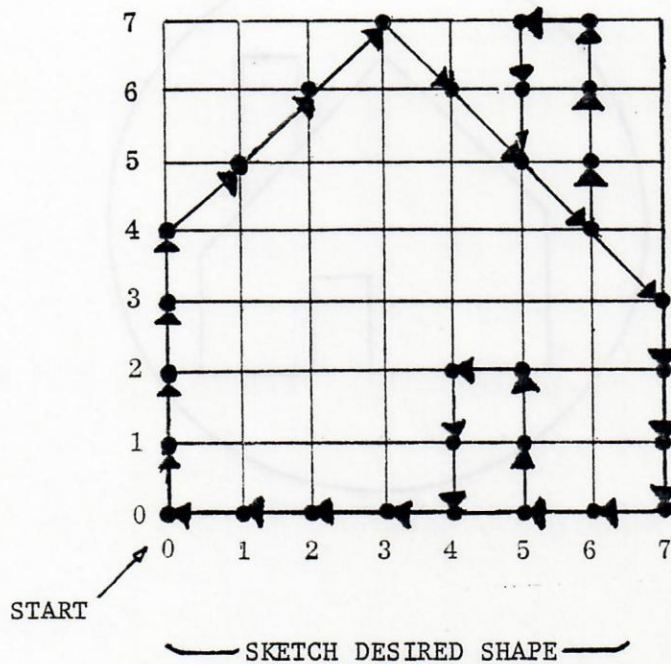
REPRODUCED CIRCUITS AND INTERFERENCE CIRCUITS

REPRODUCED 1-1-71. These are materials intended to be developed by the student and instructor with the components and the R.C. Cards available.

NUMERICAL CONTROL SYSTEM

EXERCISE 1

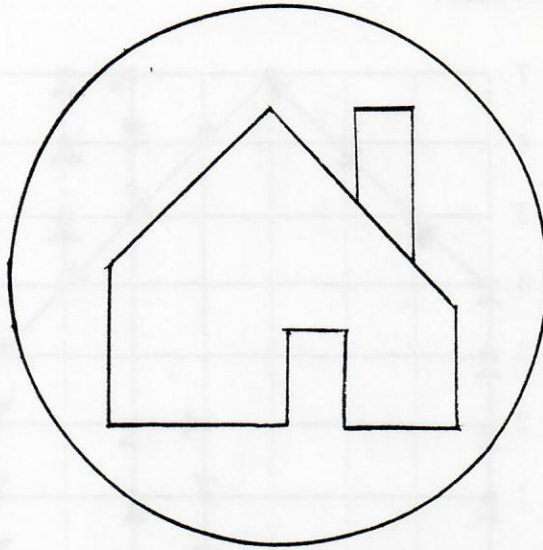
The picture selected is a house.



PROGRAM

LOCATION	WORD	LOCATION	WORD	LOCATION	WORD
00	000	13	056	26	006
01	010	14	066	27	005
02	020	15	076	30	015
03	030	16	075	31	025
04	040	17	065	32	024
05	051	20	055	33	014
06	062	21	046	34	004
07	073	22	037	35	003
10	064	23	027	36	002
11	055	24	017	37	001
12	046	25	007		

NUMERICAL CONTROL SYSTEM



COORDINATE	COORDINATE	COORDINATE	COORDINATE
100	10	100	10
200	20	200	20
300	30	300	30
400	40	400	40
500	50	500	50
600	60	600	60
700	70	700	70
800	80	800	80
900	90	900	90
1000	100	1000	100
1100	110	1100	110
1200	120	1200	120
1300	130	1300	130
1400	140	1400	140
1500	150	1500	150
1600	160	1600	160
1700	170	1700	170
1800	180	1800	180
1900	190	1900	190
2000	200	2000	200

ELECTRONIC MUSIC

NOTE: This experiment represents an application of digital electronics to the arts of music. The circuits given in this experiment are typical of the possible circuits that can be applied to this purpose. Encourage the student to experiment with various circuits and components and to try any ideas he may have on ways to generate various tones, beats, sequences etc. The possibilities for music generation with Digital Systems are still being explored by scientists and musicians throughout the world. This is an opportunity for the student to be creative and to try his own ideas.

PROCEDURE:

SETTING	SOUND
SHORT PULSE FAST ATTACK LONG DECAY	This sounds like a string being plucked as in a guitar or a harp
SHORT PULSE FAST ATTACK FAST DECAY	For very short pulses this sounds like a drum beat. When the pulse is a little long it sounds like electronic music.
LONG PULSE SLOW ATTACK FAST DECAY	This sounds like a recording being played backwards.
LONG PULSE SLOW ATTACK LONG DECAY	This sounds like a bowed instrument such as a violin.

The student should demonstrate the "music" he has produced with his system. He must generate schematics for any ideas he is testing using his music system.

EXPERIMENT NO. 17

NOTE: This experiment demonstrates an application of digital electronics to the arts of music. The circuit given in this experiment has typical of the possible circuits that can be applied to this purpose. Encourage the student to experiment with various circuits and components and to try any ideas he may have on ways to generate various tones, notes, sequences etc. The possibilities are numerous. Connections with digital systems are still being explored by scientists and engineers throughout the world. This is an opportunity for the student to be creative and to try his own ideas.

PROCEDURE:

SETTING	SOUND
SHORT PULSE FAST ATTACK LONG DECAY	This sounds like a string being plucked as in a guitar or a harp.
SHORT PULSE FAST ATTACK FAST DECAY	For very short pulses this sounds like a drum beat. When the pulse is a little long it sounds like electronic music.
LONG PULSE FAST ATTACK FAST DECAY	This sounds like a recording being played repeatedly.
LONG PULSE FAST ATTACK LONG DECAY	This sounds like a power instrument such as a violin.

The student should demonstrate the "sounds" he has produced with his system. He will generate excitement for my class as he teaches using his own system.

DIGITAL MODEM

ASCII CHARACTER ENTER TEST:

TEST	OCTAL NO.	✓
1	301	✓
2	303	✓
3	305	✓
4	207	✓
5	312	✓

PROCEDURE (RECEIVER):

WORD	SYMBOL (LETTER) (NUMBER)	TRANSMITTED ASCII CODE	RECEIVED DATA
1	F	306	306
2	O	317	317
3	R	322	322
4	SPACE	240	240
5	R	322	322
6	E	305	305
7	N	316	316
8	T	324	324
9	LINE FEED	212	212
10	Carriage Return	215	215
11	C	303	303
12	A	301	301
13	L	314	314
14	L	314	314
15	SPACE	240	240
16	A	301	301
17	J	312	312
18	A	301	301
19	X	332	332
20	!	241	241

DIGITAL MODEL

ASCII CHARACTER SET TEST:

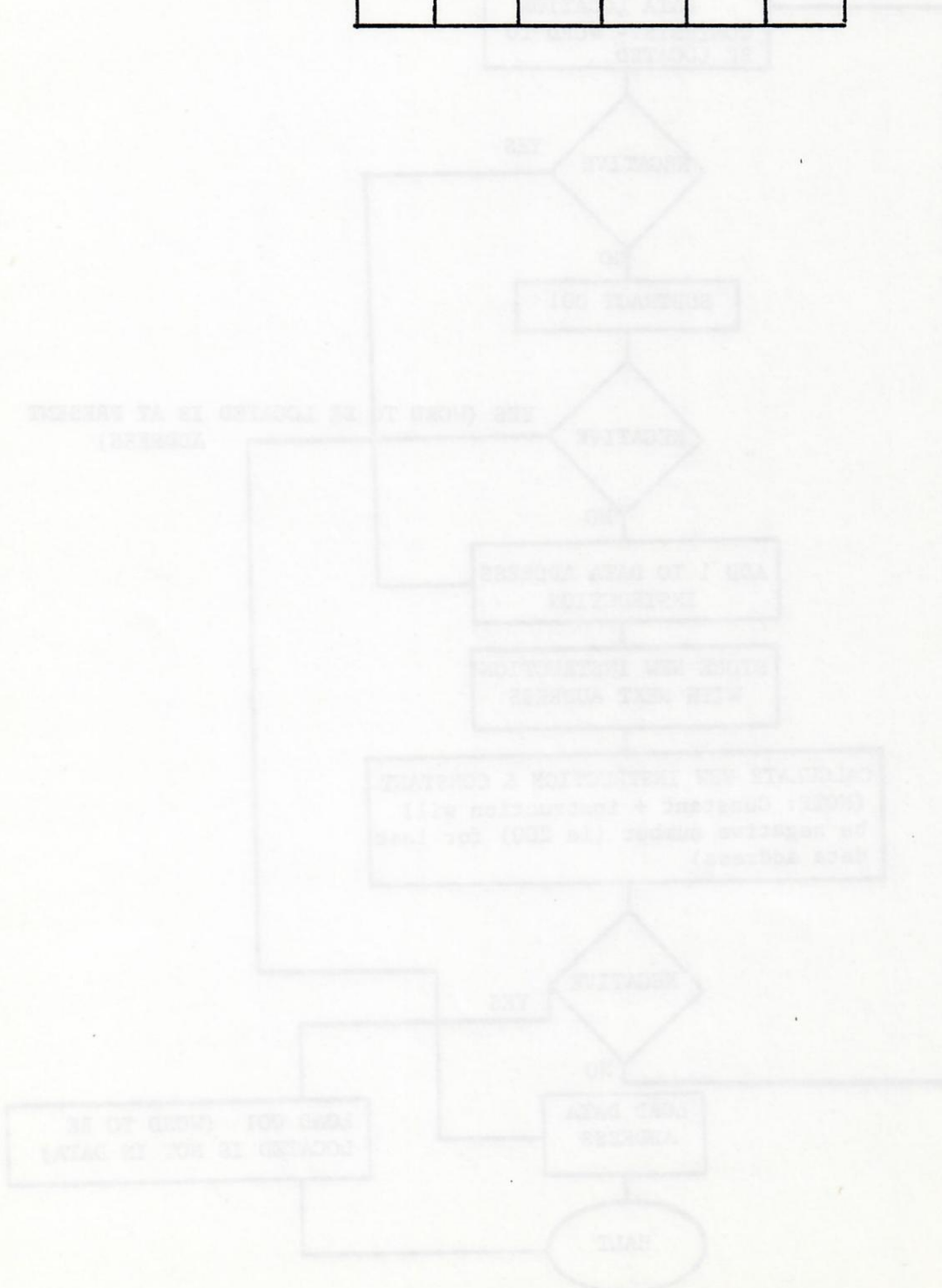
TEST	ASCII CODE	✓
1	301	✓
2	302	✓
3	303	✓
4	304	✓
5	305	✓

RECEIVED DATA:

WORD	LETTER (NUMBER)	TRANSMITTED ASCII CODE	RECEIVED DATA
1	1	301	301
2	2	302	302
3	3	303	303
4	4	304	304
5	5	305	305
6	6	306	306
7	7	307	307
8	8	308	308
9	9	309	309
10	10	310	310
11	11	311	311
12	12	312	312
13	13	313	313
14	14	314	314
15	15	315	315
16	16	316	316
17	17	317	317
18	18	318	318
19	19	319	319
20	20	320	320

ADDITIONAL PROGRAMMING METHODSSORT

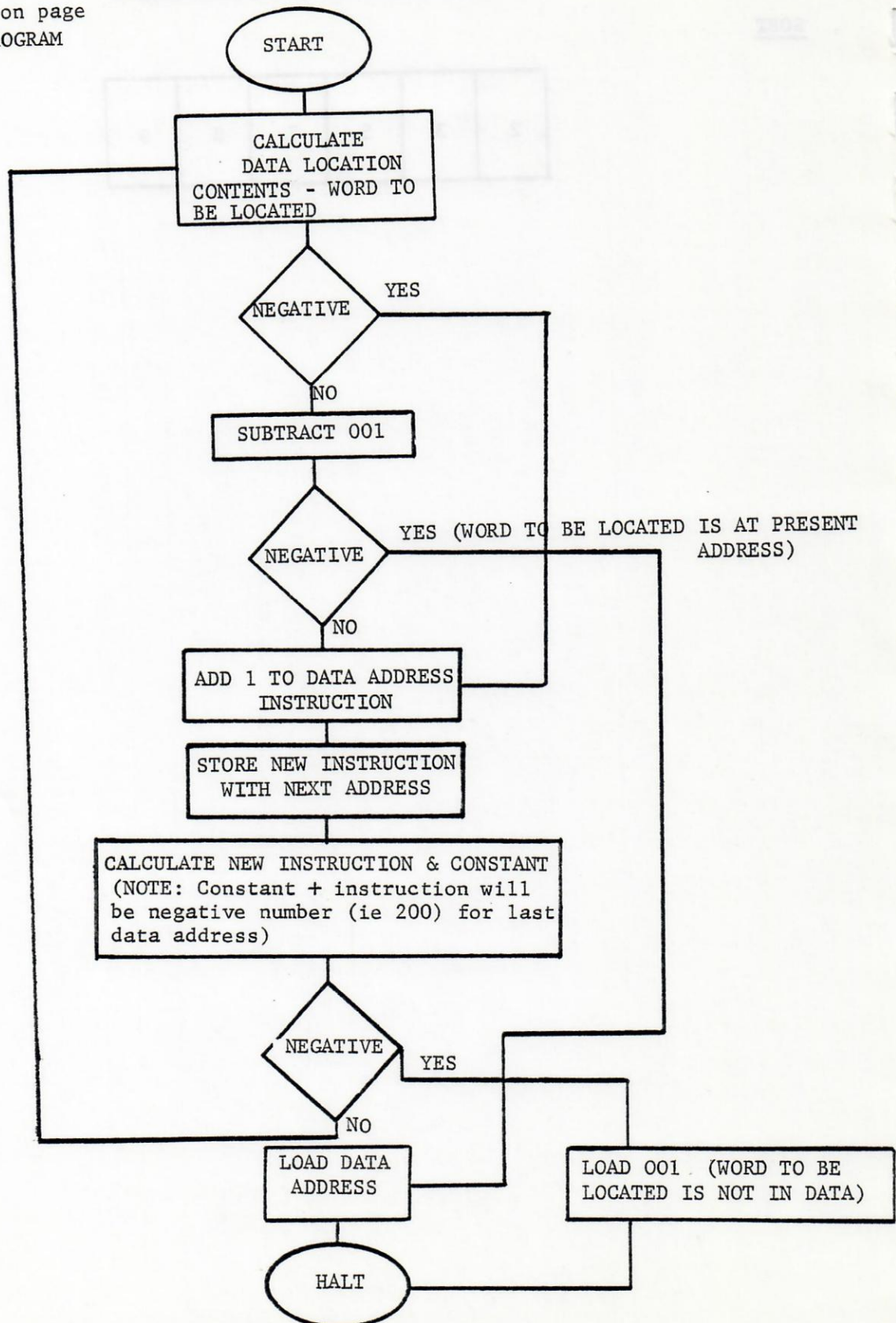
2	3	5	7	8	9
---	---	---	---	---	---



ADDITIONAL PROGRAMMING METHODS

EXERCISE:

1) The program coded on page
4 is a LOCATION PROGRAM
FLOW CHART



ADDITIONAL PROGRAMMING METHODS

TEST	WORD TO BE LOCATED (02)	LOCATION
1	HUMAN 310	106
2	IN 311	046
3	DISOLVE 304	001
4	IT 311	046
5	EVENTS 305	116
6	NECESSARY 316	146
7	OF 317	076

COMMENT

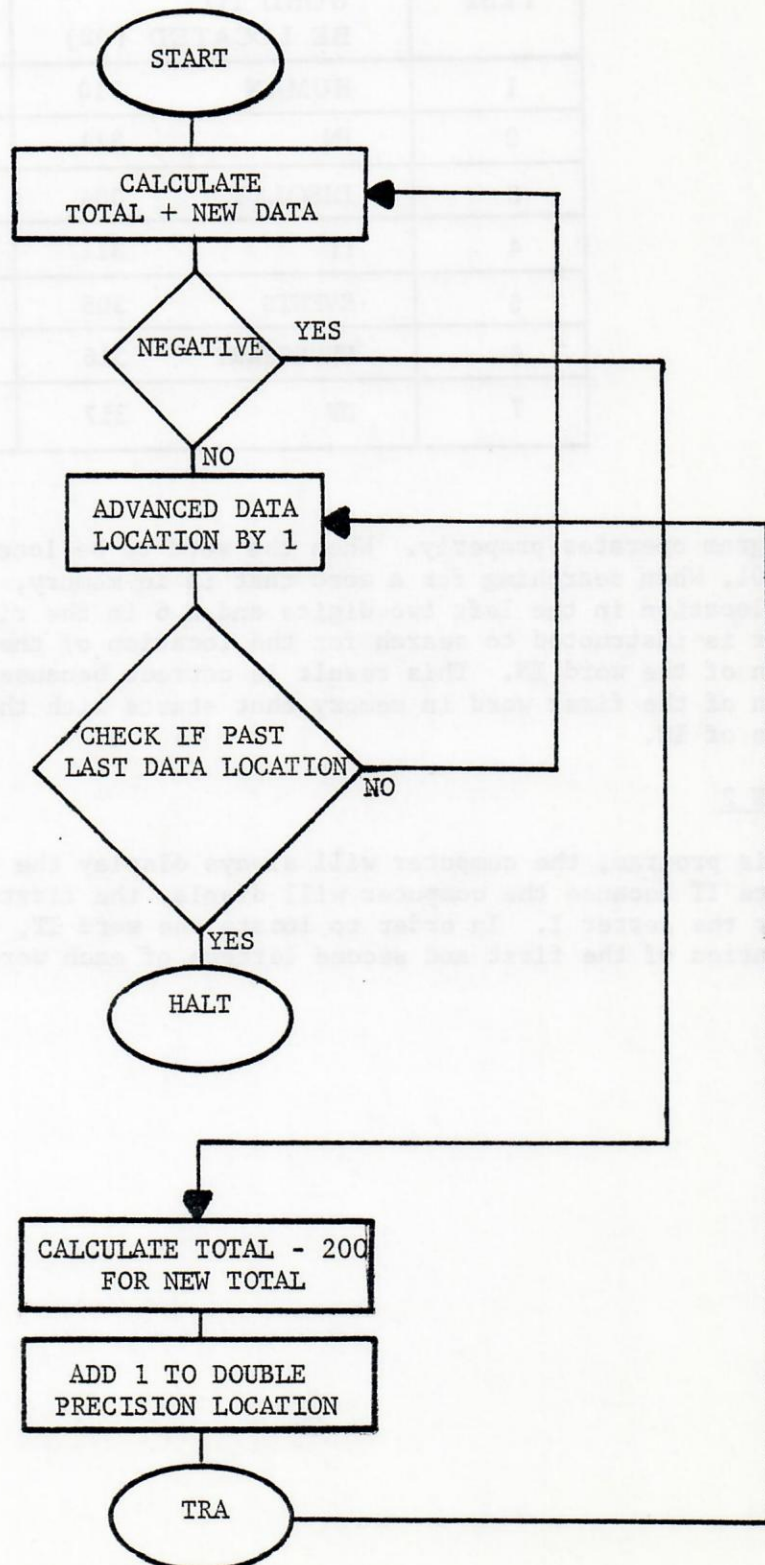
The program operates properly. When the word to be located is not in memory the display reads 001. When searching for a word that is in memory, the computer halts with the memory location in the left two digits and a 6 in the right hand digit. When the computer is instructed to search for the location of the word IT the result is the location of the word IN. This result is correct because the program displays the location of the first word in memory that starts with the letter I. This is the location of IN.

QUESTION 2

With this program, the computer will always display the location of IN when instructed to locate IT because the computer will display the first location containing the ASCII code for the letter I. In order to locate the word IT, the computer must search for the location of the first and second letters of each word.

ADDITIONAL PROGRAMMING METHODS

EXERCISE 3
DOUBLE PRECISION



FLOW CHART - DOUBLE PRECISION

ADDITIONAL PROGRAMMING METHODS

In the SYMBOLIC PROGRAM more than 32 memory locations were needed. The number of data words was reduced to six to fit into memory.

	WORD	COMMENT
	L(1) TRANS	TRANSFER AROUND DATA
L(ANS)	Ø	
	Ø	
	Ø	
	Ø	
	Ø	
	Ø	
L(DP)	Ø	OVERFLOW DOUBLE PRECISION LOCATION
L(1)	L(02) LOAD	LOAD DATA
	L(01) ADD	ADD 1st DATA (This address becomes the Answer Location)
	L(2) TON	TRANSFER ON NEGATIVE TO L(2)
	L(01) STORE	STORE IN L(ANS)
	L(3) TRANS	TRANSFER TO L(3)
L(2)	L(200) SUB	SUBTRACT OVERFLOW FROM L(ANS)
	L(01) STORE	STORE REMAINDER IN L(ANS)
	L(DP) LOAD	LOAD L(DP) OVERFLOW STORAGE
	L(001) ADD	ADD 001 TO L(DP)
	L(DP) STORE	STORE NEW L(DP)
L(3)	L(1) LOAD	LOAD OLD DATA INSTRUCTION
	L(010) ADD	ADVANCE DATA INSTRUCTION
	L(1) STORE	STORE NEW DATA INSTRUCTION
	L(C) SUB	SUBTRACT CONSTANT
	L(1) TON	TRANSFER ON NEGATIVE TO L(1)
	L(01) LOAD	LOAD ANSWER
	L(NO) HALT	HALT
	L(DP) LOAD	LOAD L(DP)

ADDITIONAL PROGRAMMING METHODS

SYMBOLIC PROGRAM (Continued)

	WORD	COMMENT
	L(NO) HALT	HALT
L(010)	Ø	
L(200)	Ø	
L(NO)	Ø	
L(001)	Ø	
L(C)	Ø	

ADDITIONAL PROGRAMMING METHODS

PROGRAM : DOUBLE PRECISION					
ADDRESS		WORD			COMMENTS
0	0	0	7	(1)	TRANSFER TO 10
0	1	0	0	0	
0	2	0	0	0	
0	3	0	0	0	DATA
0	4	0	0	0	
0	5	0	0	0	
0	6	0	0	0	
0	7	0	0	0	
1	0	0	2	(6)	LOAD L(02)
1	1	0	1	(5)	ADD L(01)
1	2	1	4	(0)	TON TO L(15)
1	3	0	1	(2)	STORE IN L(01) (ANS)
1	4	2	1	(1)	TON TO L(22)
1	5	3	4	(4)	SUBTRACT 200
1	6	0	1	(2)	STORE ANS
1	7	0	7	(6)	LOAD L(DP)
2	0	3	6	(5)	ADD 001
2	1	0	7	(2)	STORE NEW DP
2	2	1	0	(6)	LOAD L(10)
2	3	3	3	(5)	ADD 010
2	4	1	0	(2)	STORE INSTRUCTION WITH NEW ADDRESS
2	5	3	7	(4)	SUBTRACT CONSTANT
2	6	0	7	(0)	TON TO L(10)
2	7	0	1	(6)	LOAD L(01) (ANSWER)
3	0	3	5	(7)	HALT
3	1	0	7	(6)	LOAD L(DP)
3	2	3	5	(7)	HALT
3	3	0	1	0	010
3	4	2	0	0	200
3	5	0	0	0	L(NO)
3	6	0	0	1	001
3	7	0	7	0	CONSTANT

ADDITIONAL PROGRAMMING METHODS

OPERATOR INSTRUCTIONS

LOAD DATA INTO LOCATIONS 01 THROUGH 06
LOAD 000 INTO LOCATION 07
LOAD 026 INTO 10

START AT 00
AUTOMATIC HALT
DISPLAY SHOWS L(ANS) (LEAST SIGNIFICANT ANS)
PRESS START
AUTOMATIC HALT
DISPLAY SHOW L(DP) DOUBLE PRECISION LOCATION WITH OVERFLOW

PROGRAM TEST

TEST	1	2
Ø1	010	100
Ø2	100	162
Ø3	100	032
Ø4	030	102
Ø5	004	101
Ø6	003	050
ANS	047	167
DP	001	002
CHECK	✓	✓

TEST 1

010	
100	
100	L(ANS) = 047
030	L(DP) = 001
004	
003	
247	

TEST 2

100	
162	
032	L(ANS) = 167
102	L(DP) = 002
101	
050	
567	

ADDITIONAL PROGRAMMING METHODS

EXERCISE 4

Triple Precision will allow 3 memory locations to be used for each number. The count sequence would be:

000	000	000
000	000	001
.	.	.
.	.	.
.	.	.
000	000	177
000	001	000
.	.	.
.	.	.
.	.	.
000	177	177
.	.	.
.	.	.
.	.	.
001	000	000
.	.	.
.	.	.
.	.	.
377	177	177

The maximum number stored would be $7 + 7 + 8$ BITS or 22 BITS for a maximum of $2^{22} = 753,151$

EXERCISE 5

The process of multiplication can easily cause overflow because of the large numbers that can be generated. Double Precision offers a method of handling these large numbers

REAL - TIME TIMING

N COUNT	TIME (SECONDS)
100	23

ADDITIONAL PROGRAMMING METHODS

TEST	CALCULATED		MEASURED
	N COUNT IN (06)	TIME (SECONDS)	ELAPSED TIME (SECONDS)
1	100		23
2	177 not high enough	60	--
3	74	22	21
4	153	38.4	36
5	001 (minimum)	23/100	not measurable
6	177 (maximum)	45.6	44
7	34	10	10

In Test #2, the maximum time count of 177 will result in a time of less than 60 seconds.

Calculations Required (Note: Calculations are done in decimal then converted to octal)
TEST 3

100 OCTAL = 64 DECIMAL therefore 64 counts requires 23 seconds elapsed time.

$$\frac{23}{64} = \frac{22}{N}, N = \frac{22(64)}{23} = 61 \text{ DECIMAL}$$

61 DECIMAL = 74 OCTAL

TEST 4

153 OCTAL = 107 DECIMAL

$$\frac{23}{64} = \frac{T}{107}, T = \frac{23(107)}{64} = 38.4 \text{ SECONDS}$$

TEST 6

177 OCTAL = 127 DECIMAL

$$\frac{T}{127} = \frac{23}{64}, T = \frac{23(127)}{64} = 45.6 \text{ SECONDS}$$

TEST 7

T = 10 SECOND

$$\frac{23}{64} = \frac{10}{N}, N = \frac{10(64)}{23} = 28 \text{ DECIMAL}$$

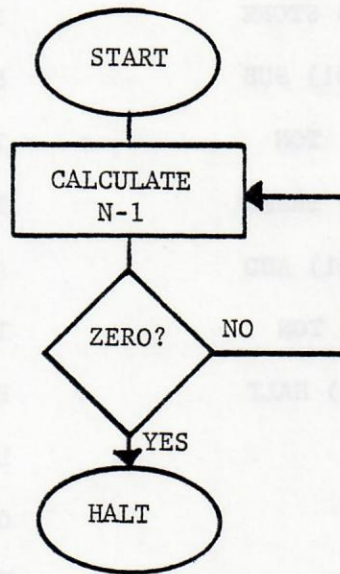
28 DECIMAL = 34 OCTAL = N

ADDITIONAL PROGRAMMING METHODS

The measured and calculated results are within 10% of each other, which is approximately the limits of measurement with a wrist watch second hand.

EXERCISE 6

FLOW CHART -EXPANDED TIMING



ADDITIONAL PROGRAMMING METHODS

SYMBOLIC PROGRAM : EXPANDED TIMING

	WORD	COMMENT
L(4)	L(2) LOAD	LOAD N
	L(001) SUB	SUBTRACT 1
	L(2) STORE	STORE NEW N
	L(001) SUB	SUBTRACT 1
	L(3) TON	TRANSFER ON NEGATIVE TO L(3)
	L(4) TRANS	TRANSFER TO L(4)
L(3)	L(001) ADD	ADD 001
	L(4) TON	TRANSFER ON NEGATIVE TO L(4)
	L(NO) HALT	HALT
L(NO)	000	L(NO)
L(001)	001	001
L(2)	Ø	NUMBER OF CYCLES

ADDITIONAL PROGRAMMING METHODS

PROGRAM :						
ADDRESS		WORD			COMMENTS	
0	0	1	3	(6)	LOAD N	
0	1	1	2	(4)	SUB 1	
0	2	1	3	(2)	STORE NEW N	
0	3	1	2	(4)	SUB 1	
0	4	0	5	(0)	TON TO 06	
0	5	3	7	(1)	TRA TO 00	
0	6	1	2	(5)	ADD 1	
0	7	3	7	(0)	TON TO 00	
1	0	1	1	(7)	HALT	
1	1	0	0	0	ZERO	
1	2	0	0	1	1	
1	3	0	0	0	N	

OPERATOR INSTRUCTIONS

LOAD N IN LOCATION 13
START AT 00
AUTOMATIC HALT

CALIBRATION

LOAD N = 100 (OCTAL) = 64 (DECIMAL)
RUN PROGRAM: TIME = 43 SECONDS

ADDITIONAL PROGRAMMING METHODS

TEST	CALCULATED		MEASURED
	N COUNT IN 13	TIME SECONDS	ELAPSED TIME (SECONDS)
1	100	43	43
2	200	86	87
3	255	118	122
4	131	60	60

TEST 1

CALIBRATION - OCTAL 100 - DECIMAL 64
TIME IS MEASURED - 43 SECONDS ELAPSED TIME

TEST 2

N = 200 (OCTAL) = 128 DECIMAL

$$\frac{T}{128} = \frac{43}{64}, \quad T = \frac{43(128)}{64} = 86 \text{ SECONDS}$$

TEST 3

N = 255 (OCTAL) = 175 DECIMAL

$$\frac{T}{175} = \frac{43}{64}, \quad T = \frac{43(175)}{64} = 118$$

TEST 4

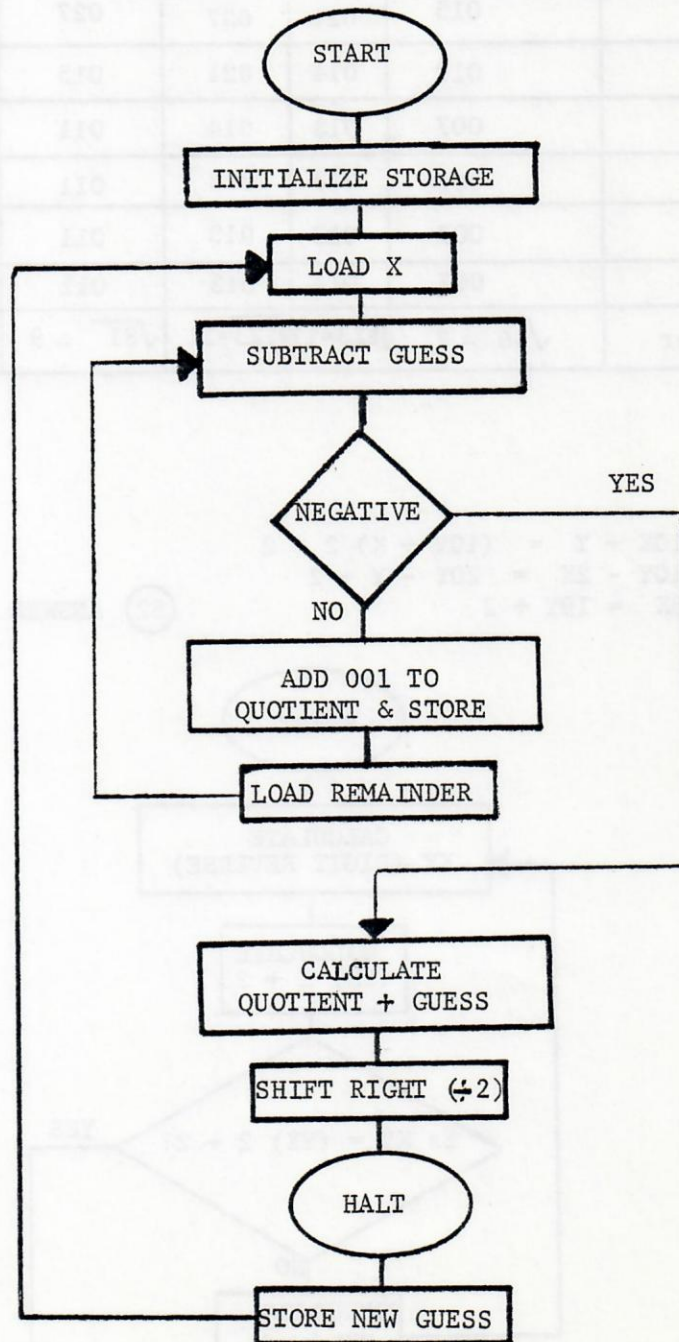
T = 60 SECONDS

$$\frac{60}{N} = \frac{43}{64}, \quad N = \frac{60(64)}{43} = 89 \text{ (DECIMAL)}$$

N = 89 DECIMAL = 131 (OCTAL)

ADDITIONAL PROGRAMMING METHODSSQUARE ROOT PROGRAM

The FLOWCHART is given below. Note that this program must be started at STEP 02. This assures that the program will not mistake the number X as an instruction and modify the program unintentionally.

EXERCISE 7

ADDITIONAL PROGRAMMING METHODS

FIND $\sqrt{N}$	example 1	2	3	4	5	6
N octal	061	173	173	121	100	142
enter first guess	061	100	173	121	100	142
computer 1	032	040	076	056	040	061
2	015	021	037	027	021	031
3	010	014	021	015	012	016
4	007	013	014	011	010	011
5	007	013	013	011	010	011
6	007	013	013	011	010	011
7	007	013	013	011	010	011
correct answer	$\sqrt{49} = 7$	$\sqrt{123}=11$	$\sqrt{123}=11$	$\sqrt{81} = 9$	$\sqrt{64}=8$	$\sqrt{98}=9$

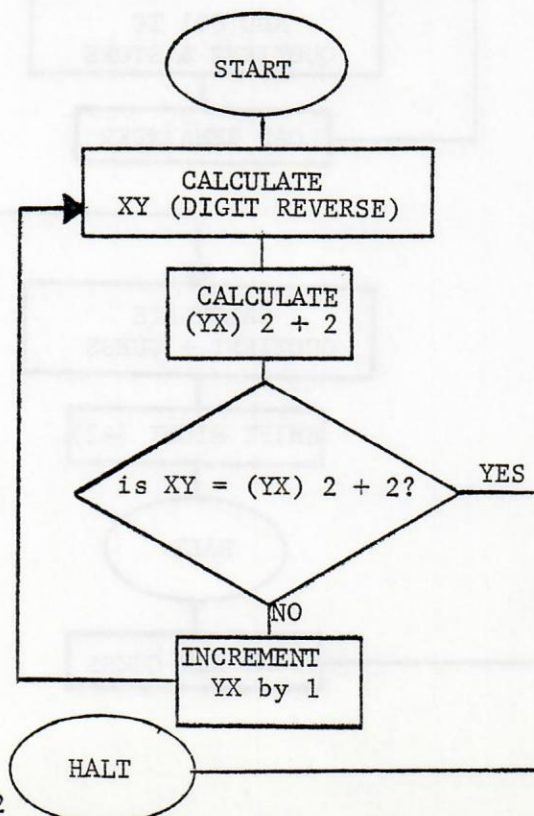
EXERCISE 8Number = $10Y + X$

$$10X + Y = (10Y + X) 2 + 2$$

$$10Y - 2X = 20Y - Y + 2$$

$$8X = 19Y + 2$$

(52) ANSWER



$$10Y + X = (10X + Y) 2 + 2$$

INPUT-OUTPUT INTERFACEPROCEDURE:Exp. 20-1 (0.5 Volt Resolution Display)

F	G	H	J	K	DISPLAY
8	4	2	1	.5	
0	0	1	1	1	3.5
1	0	0	1	1	9.5
0	1	0	1	1	5.5
0	1	0	0	0	4.0
0	1	1	0	1	6.5

Exp. 20-2 (0.25 Volt Resolution Display)

F	G	H	J	K	DISPLAY
4	2	1	.5	.25	
0	0	1	1	1	1.75
0	1	0	1	0	2.50
0	0	1	0	0	1.00
1	1	0	1	0	6.50
1	1	1	0	0	7.00
1	0	1	1	1	5.75

INPUT-OUTPUT INTERFACEExp. 20-3 (Digital Voltmeter with Display)

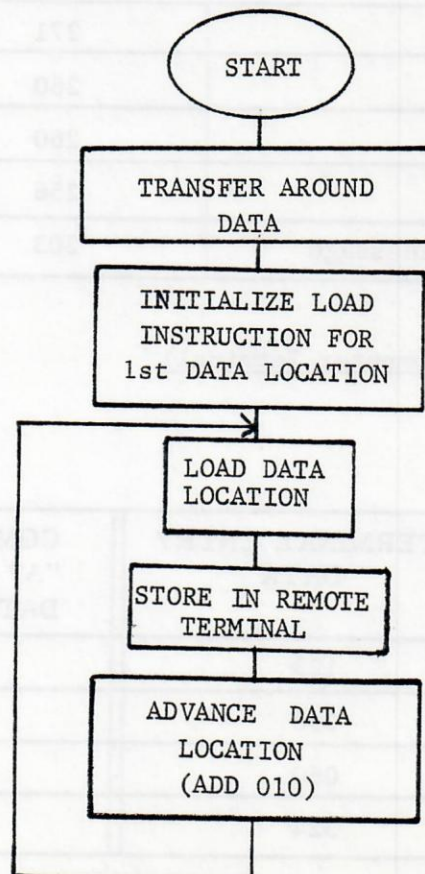
SET VOLTAGE	DISPLAY READING
0	0.00
0.25	0.25
0.50	0.50
0.75	0.75
1.00	1.00
1.25	1.25
1.50	1.50
1.75	1.75
2.0	2.00
2.25	2.25
2.50	2.50
2.75	2.75
3.0	3.00
3.25	3.25
3.50	3.50
3.75	3.75
cannot increase voltage above 3.75	

INPUT-OUTPUT INTERFACEREMOTE OUTPUT DISPLAY:

Note: The computer is operated in One Cycle Mode, observe the data appear on the Display of the Remote Terminal as the computer advances through the program.

WORD	DATA
1	303
2	305
3	323

EXERCISE 1
FLOW CHART



INPUT-OUTPUT INTERFACEPROCEDURE: (Remote Output Display Terminal)DATA TABLE

WORD	ALPHA NUMERIC	ASCII CODE	CHECK
01	Start Message	201	201
02	Line Feed	212	212
03	Carriage Return	215	215
04	E	305	305
05	D	304	304
06	Space	240	240
07	L	314	314
10	A	301	301
11	B	302	302
12	Space	240	240
13	9	271	271
14	0	260	260
15	0	260	260
16	Period	256	256
17	End of Message	203	203

PROCEDURE: (Remote Input-Output Computer Terminal)

TEST	TERMINAL ENTRY DATA	COMPUTER "A" REGISTER DATA
1	123	123
2	050	050
3	060	060
4	324	324

INPUT-OUTPUT INTERFACE

REMOTE TERMINAL PROGRAM:

PROGRAM TEST: $X = B + 2A$

A	B	X
007	002	020

TEST	A	B	Computer Answer $B + 2A$
1	002	010	014
2	023	111	157
3	002	370	374

EXERCISE 2PROGRAM $X = B + \frac{A}{2}$ REMOTE PROGRAM FOR $X = B + \frac{A}{2}$

ADDRESS	WORD	COMMENT
25	33(6)	LOAD A
26	00(3)	SHIFT RIGHT
27	32(5)	ADD B
30	37(2)	STORE
31	37(1)	TRANSFER TO L(00)
32	Ø	B
33	Ø	A
34	37(6)	

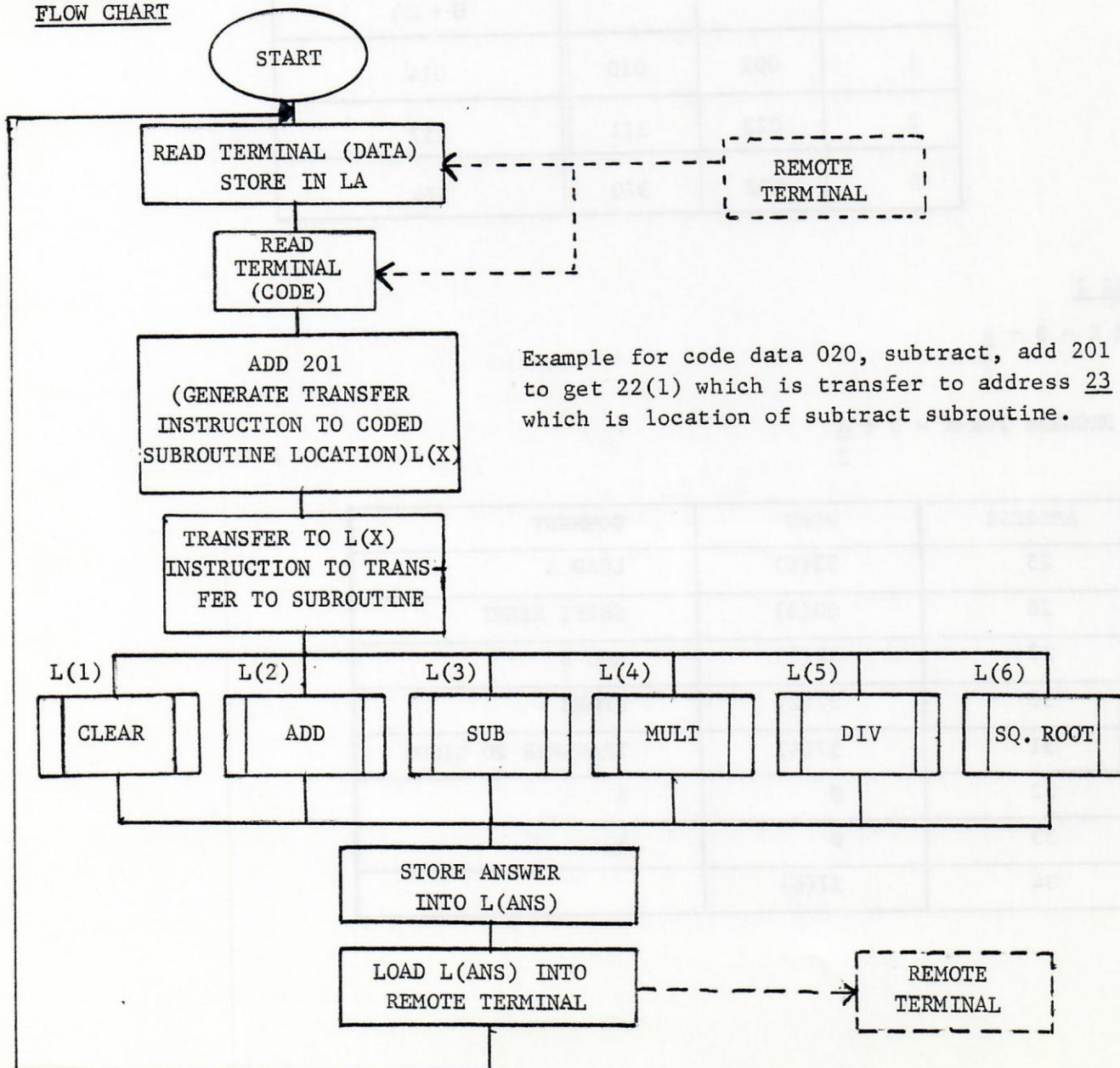
INPUT-OUTPUT INTERFACE

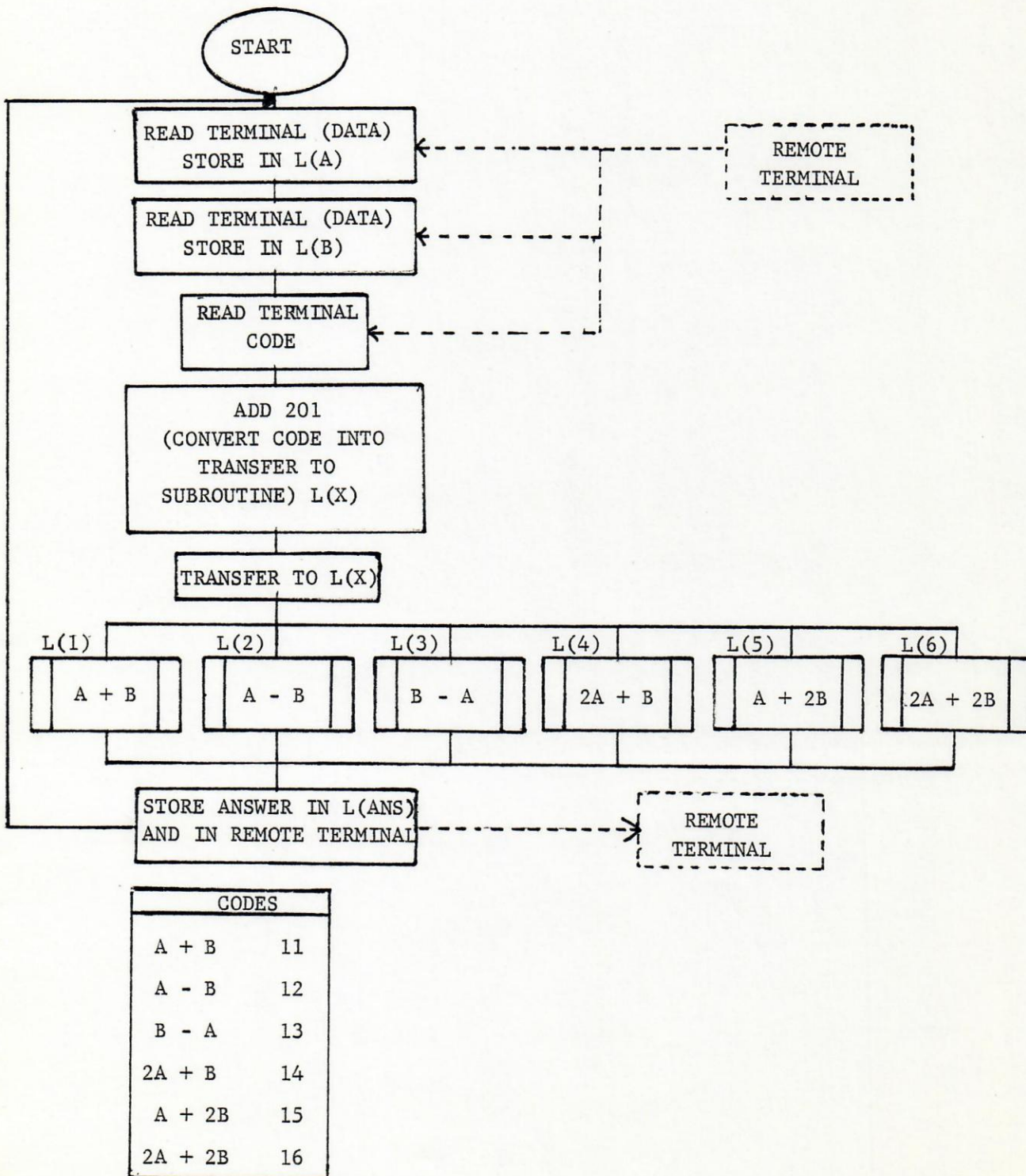
PROGRAM TEST: $X = B + \frac{A}{2}$

Note: Use only even numbers for A; (if an odd number is used, the extra 1 is end around shifted and the answer will be 200 more than expected.)

TEST	A	B	X
1	006	004	007
2	024	105	117
3	010	200	204

EXERCISE 3
FLOW CHART



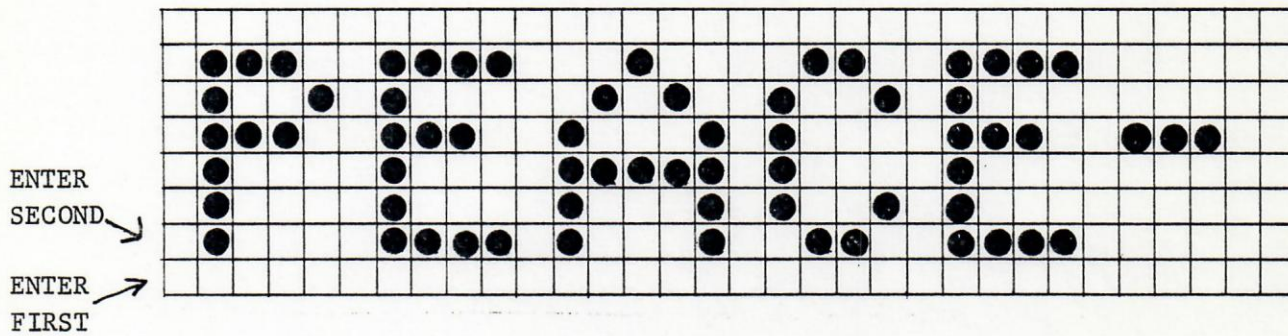
INPUT-OUTPUT INTERFACEEXERCISE 4
FLOW CHART

ALPHANUMERIC DISPLAY SYSTEM

PROCEDURE:

4. Are the proper letters display? YES

5.

6. Are the proper letters display? YESINSTRUCTORS NOTE:

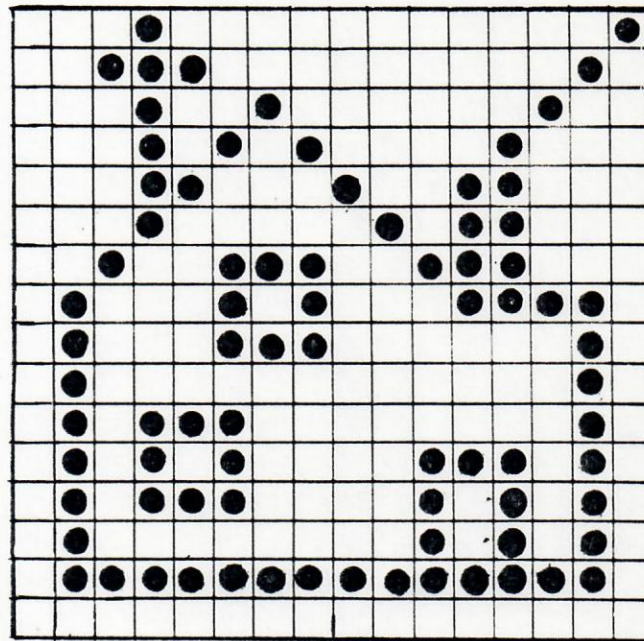
When programming the Alphanumeric Display system it is advisable to leave the top and bottom rows of dots blank and to leave at least 2 empty columns at the end of the display. This assures a neat and readable display.

GRAPHICS CRT DISPLAY

PROCEDURE:

3. Run the graphics display. Does the proper pattern appear on the oscilloscope?
YES

5.



ENTER SECOND →

ENTER FIRST →

6. Does the proper pattern appear? YES

INSTRUCTORS NOTE:

The patterns produced with this circuit are composed of individual dots. The same House diagram produced in Experiment 16 was generated by a moving dot following the pattern on the screen.

MORSE CODE KEYS

PROCEDURE:

4. Does the Audio output correspond to the Morse Code pattern recorded in memory? YES

INSTRUCTORS NOTE:

The Morse Code generated by this system is very precise and highly accurate. Also it can be transmitted very rapidly. Compare the morse code output of the machine with the same pattern being sent by a person manipulating a telegraph key. The difference in the "sound" of the message will be noticeable.

HOUSE CODE SYSTEM

REFERENCE:

4. Does the radio output correspond to the House Code pattern recorded in
memory? YES

INSTRUCTIONS:

The House Code generated by this system is very precise and highly accurate. Also
it can be transmitted very rapidly. Compare the words code output of the machine
with the words pattern being sent by a person transmitting a telegraph key. The
distinction in the "sound" of the message will be noticeable.

AUTOMATIC KEYBOARD ENTRYPROCEDURE:

ENTER DATA	CHECK
1 2 3	123
2 4 7	247
3 0 1	301
0 2 4	024
0 0 6	006
1 2 4	124
3 2 1	321

Does this system operate properly? YES

INSTRUCTORS NOTE:

This automatic data entry system can be used with the digital computer experiment to facilitate the entry of programs into the machine.

This may be done by using a second 700 to connect the automatic entry circuit or by using the Ed-Lab 101 Occupational Skills Trainer and plugging the finished circuit into the P.C. card socket on the Ed-Lab 900.

ALPHABETIC KEYBOARD TEST

PROCEEDING:

ENTER DATA	CHECK
1 2 3	1 2 3
4 5 6	4 5 6
7 8 9	7 8 9
0 1 2	0 1 2
3 4 5	3 4 5
6 7 8	6 7 8
9 0 1	9 0 1
2 3 4	2 3 4

Does this system operate properly? YES

INSTRUCTIONS:

This automatic data entry system can be used with the digital computer system to facilitate the entry of answers into the machine.

This may be done by using a special key to connect the automatic entry circuit or by using the 24-key 101 International Serial Printer and plugging the 101 cable into the 24-key socket on the 24-key 101.

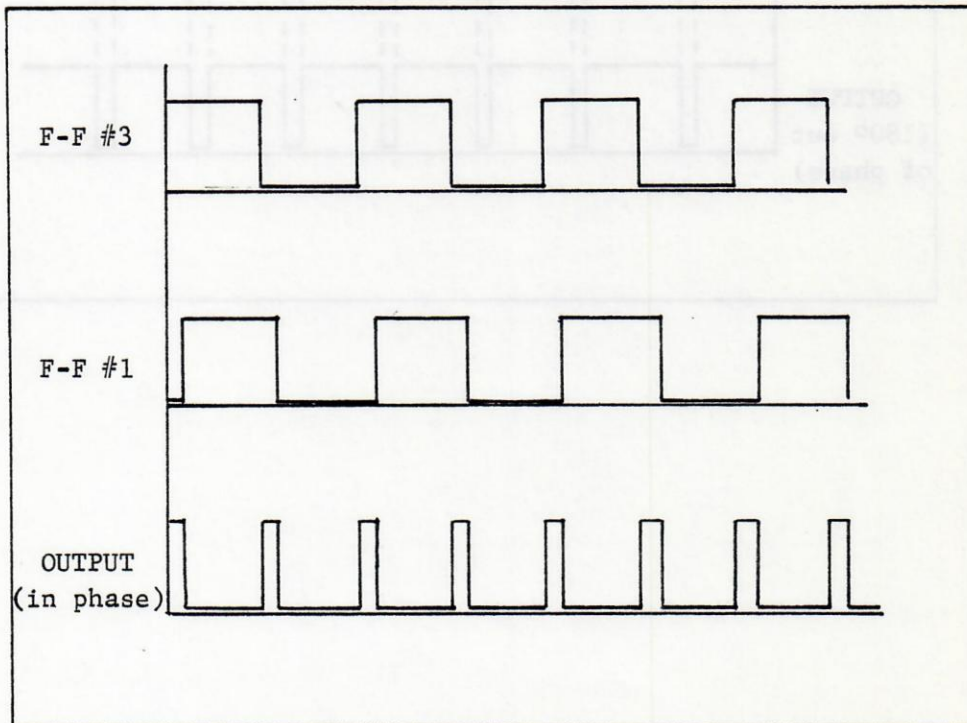
DIGITAL FREQUENCY SYNTHESIZERExp. 25-1 Digital Frequency Synthesizer

PROCEDURE: (FREQUENCY SYNTHESIZER)

2. The frequency of the tone generated by the Digital Frequency Synthesizer varies with the number set on Switches G through K. The higher the number set on the Switches, the lower the frequency of the tone.

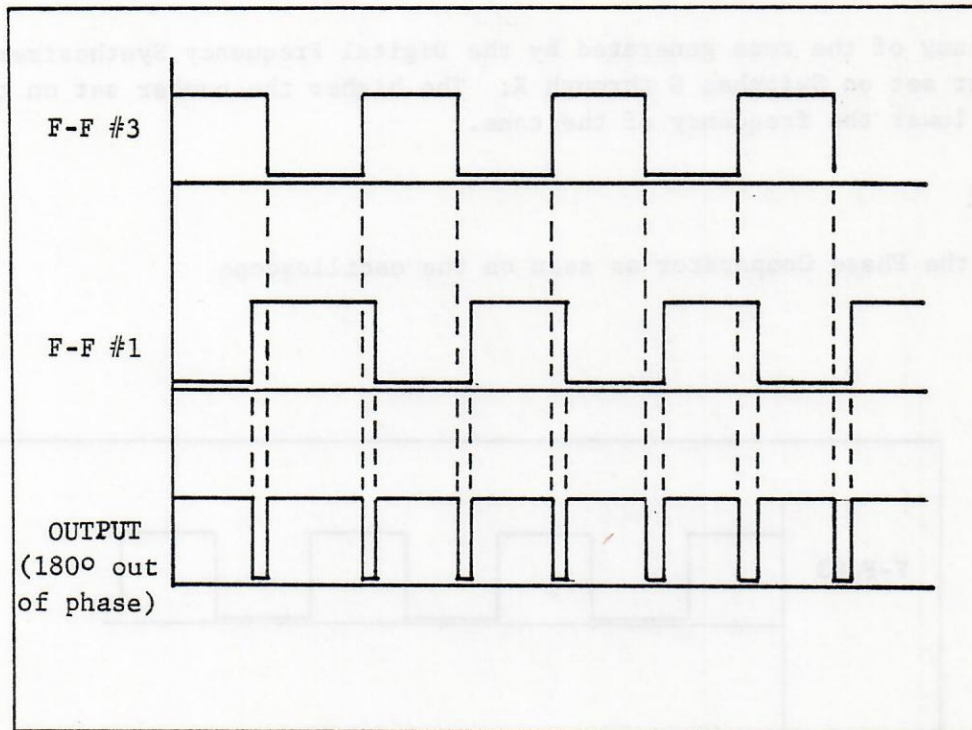
PHASE DETECTOR

3. Sketch of the Phase Comparator as seen on the oscilloscope



DIGITAL FREQUENCY SYNTHESIZER

Note: If the signals are 180° out of phase, the signals would look like this:



DIGITAL FREQUENCY SYNTHESIZERFREQUENCY COMPARATOR

S	$N = (16 - S)$	$F_o = NF_R$	F_o (SIGNAL GENERATOR)
1111 = 15	1	2000	2000 Hz (SET)
1110 = 14	2	1000	1000 Hz
1101 = 13	3	667	670 Hz
1100 = 12	4	500	500 Hz
1011 = 11	5	400	400 Hz
1010 = 10	6	333	340 Hz
1001 = 9	7	286	280 Hz
1000 = 8	8	250	255 Hz
0111 = 7	9	222	228 Hz
0110 = 6	10	200	202 Hz
0101 = 5	11	182	185 Hz
0100 = 4	12	167	168 Hz
0011 = 3	13	154	154 Hz
0010 = 2	14	143	143 Hz
0001 = 1	15	133	132 Hz
0000 = 0	16	125	125 Hz

DIGITAL FREQUENCY SYNTHESIZER

VCO OSCILLATOR

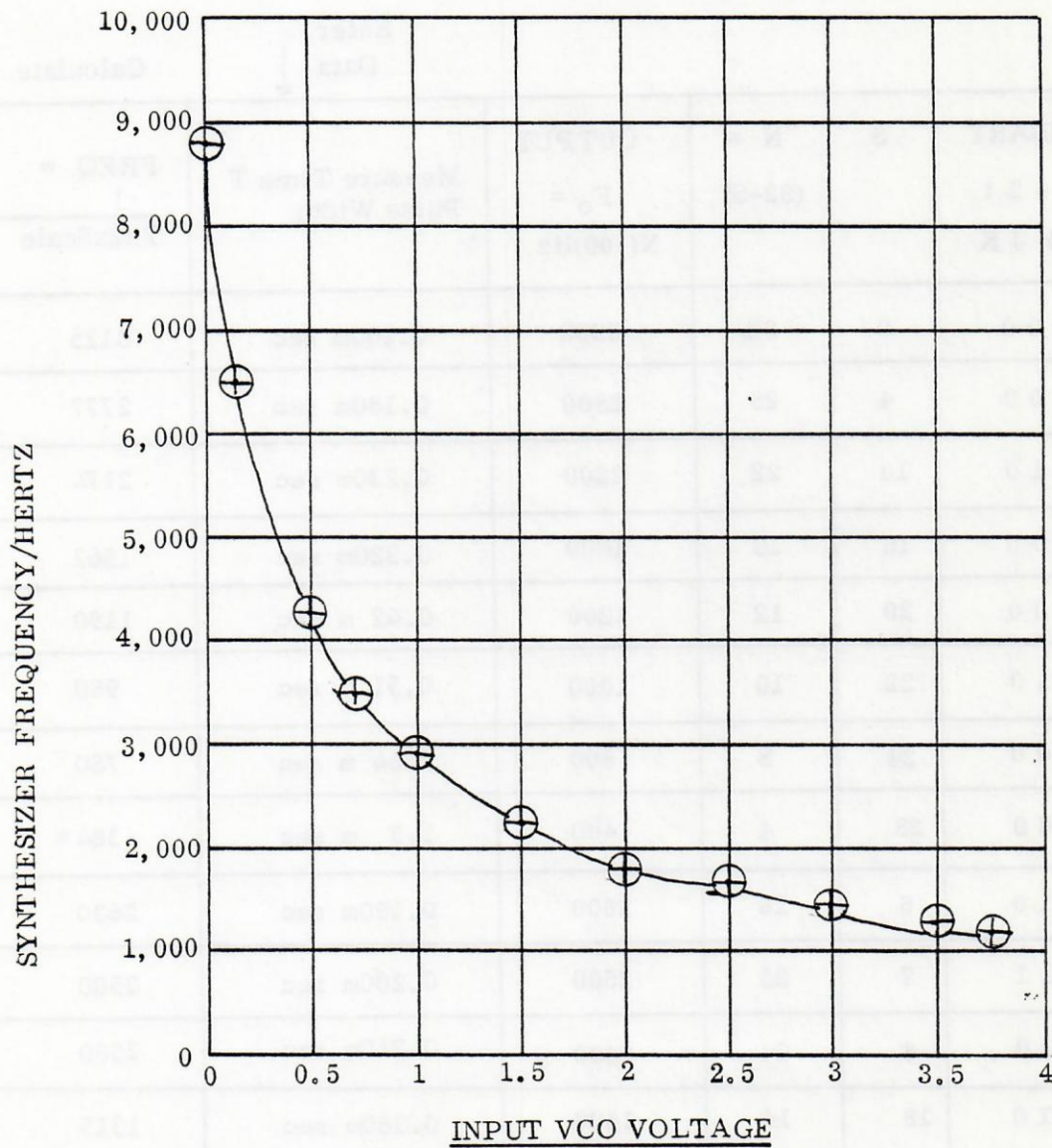
2.

ADJUST SIGNAL GENERATOR FREQUENCY

VOLTAGE SET X POT	FREQUENCY AT BALANCE
0	8,900
0.2	6,500
0.5	4,300
0.75	3,450
1.0	2,960
1.5	2,320
2.0	1,900
2.5	1,670
3.0	1,490
3.5	1,320
(Maximum)	1,290

DIGITAL FREQUENCY SYNTHESIZER

3. GRAPHICAL RESULTS of VCO INPUT VOLTAGE versus FREQUENCY CURVE and data.



What is the range of operation?

VOLTAGE FROM 0 to 3.75 Volts.FREQUENCY FROM 8,900 to 1,290 Hz.

DIGITAL FREQUENCY SYNTHESIZERPLL OPEN LOOP

4.

				Enter Data	Calculate
S BINARY 16 8 4 2 1 F G H J K	S	N = (32-S)	OUTPUT $F_o =$ N(100)Hz	Measure Time T Pulse Width	FREQ = $\frac{1}{T \times 2 \times \text{Scale}}$
0 0 0 0 0	0	32	3200	0.160m sec	3125
0 0 1 0 0	4	28	2800	0.180m sec	2777
0 1 0 1 0	10	22	2200	0.230m sec	2174
1 0 0 0 0	16	16	1600	0.320m sec	1562
1 0 1 0 0	20	12	1200	0.42 m sec	1190
1 0 1 1 0	22	10	1000	0.51 m sec	980
1 1 0 0 0	24	8	800	0.64 m sec	780
1 1 1 0 0	28	4	400	1.3 m sec	384 *
0 0 1 1 0	6	26	2600	0.190m sec	2630
0 0 1 1 1	7	25	2500	0.200m sec	2500
0 1 0 0	8	24	2400	0.210m sec	2380
1 0 0 1 0	18	14	1400	0.380m sec	1315
1 1 0 0 1	25	7	700	0.740m sec	676

* NOTE: Generator does not achieve 400Hz. Add Flip-Flop #5 to Flip-Flop #4 to divide frequency by 2.

DIGITAL FREQUENCY SYNTHESIZERPLL CLOSED LOOP

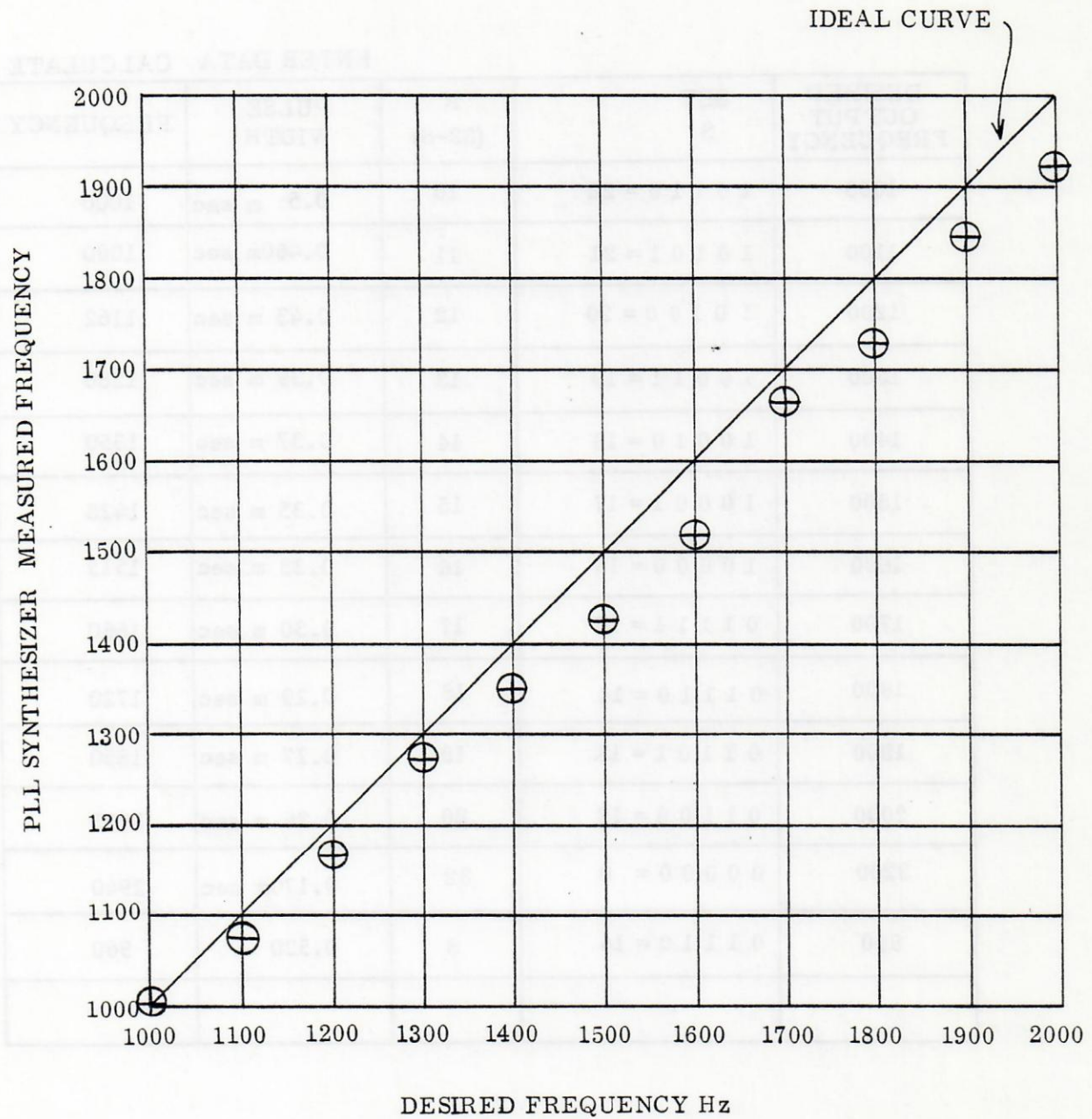
5.

ENTER DATA CALCULATE

DESIRED OUTPUT FREQUENCY	SET S	N (32-S)	PULSE WIDTH	FREQUENCY
1000	1 0 1 1 0 = 22	10	0.5 m sec	1000
1100	1 0 1 0 1 = 21	11	0.460m sec	1080
1200	1 0 1 0 0 = 20	12	0.43 m sec	1162
1300	1 0 0 1 1 = 19	13	0.39 m sec	1280
1400	1 0 0 1 0 = 18	14	0.37 m sec	1350
1500	1 0 0 0 1 = 17	15	0.35 m sec	1428
1600	1 0 0 0 0 = 16	16	0.33 m sec	1515
1700	0 1 1 1 1 = 15	17	0.30 m sec	1660
1800	0 1 1 1 0 = 14	18	0.29 m sec	1720
1900	0 1 1 0 1 = 13	19	0.27 m sec	1850
2000	0 1 1 0 0 = 12	20	0.26 m sec	1923
3200	0 0 0 0 0 = 0	32	0.170m sec	2940
800	0 1 1 1 0 = 14	8	0.520	960

DIGITAL FREQUENCY SYNTHESIZER

6.



DIGITAL FREQUENCY SYNTHESIZER

7. The operating range of the Phase locked loop, as measured experimentally, extends above 3200 Hz, the maximum frequency attainable. The low limit sums to be 960 Hz, since the measurement made for 800 Hz results in an output frequency of 960 Hz.

CLAIMS

1. The operating range of the above described lock, as described experimentally, extends above 1000 Hz, the maximum frequency available. The low limit must be at 500 Hz, since the experiment made for 500 Hz results in an almost frequency of 500 Hz.

